21-S3-C9484/C9488/F9488-092003

USER'S MANUAL

S3C9484/C9488/F9488
8-bit CMOS
Microcontroller
Revision 1

SAMSUNG ELECTRONICS

1 PRODUCT OVERVIEW

S3C9-SERIES MICROCONTROLLERS

Samsung's SAM88RCRI family of 8-bit single-chip CMOS microcontrollers offers a fast and efficient CPU, a wide range of integrated peripherals, and various mask-programmable ROM sizes.

A address/data bus architecture and a large number of bit-configurable I/O ports provide a flexible programming environment for applications with varied memory and I/O requirements. Timer/counters with selectable operating modes are included to support real-time operations.

S3C9484/C9488/F9488 MICROCONTROLLER

The S3C9484/C9488/F9488 single-chip CMOS microcontrollers are fabricated using the highly advanced CMOS process technology based on Samsung's latest CPU architecture.

The S3C9484 is a microcontroller with a 4K-byte mask-programmable ROM embedded.
The S3C9488 is a microcontroller with a 8K-byte mask-programmable ROM embedded.
The S3F9488 is a microcontroller with a 8K-byte multi time programmable ROM embedded.

Using a proven modular design approach, Samsung engineers have successfully developed the S3C9484/C9488/F9488 by integrating the following peripheral modules with the powerful SAM88 RCRI core:

— Five configurable I/O ports (38 pins) with 8-pin LED direct drive and LCD display
— Ten interrupt sources with one vector and one interrupt level
— One watchdog timer function with two source clock (Basic Timer overflow and internal RC oscillator)
— One 8-bit basic timer for oscillation stabilization
— Watch timer for real time clock
— Two 8-bit timer/counter with time interval, PWM, and Capture mode
— Analog to digital converter with 9 input channels and 10-bit resolution
— One asynchronous UART

The S3C9484/C9488/F9488 microcontroller is ideal for use in a wide range of home applications requiring simple timer/counter, ADC, LED or LCD display with ADC application, etc. They are currently available in 32-pin SOP/SDIP, 42-pin SDIP and 44-pin QFP package.

MTP

The S3F9488 has on-chip 8-Kbyte multi time programmable (MTP) ROM instead of masked ROM. The S3F9488 is fully compatible to the S3C9488, in function, in D.C. electrical characteristics and in pin configuration.

FEATURES

CPU

- SAM88RCRI CPU core

Memory

- 208-byte general purpose register (RAM)
- 4/8-Kbyte internal mask program memory
- 8-Kbyte internal multi time program memory (S3F9488)

Oscillation Sources

- Crystal, Ceramic, RC
- CPU clock divider (1/1, 1/2, 1/8, 1/16)

Instruction Set

- 41 instructions
- IDLE and STOP instructions added for power-down modes

Instruction Execution Time

- 500 ns at 8-MHz f_{OSC} (minimum)

Interrupts

- 10 interrupt sources with one vector / one level

I/O Ports

- Total 38 bit-programmable pins (44QFP)
  Total 36 bit-programmable pins (42SDIP)
  Total 26 bit-programmable pins (32SDIP/32SOP)

Basic Timer

- One programmable 8-bit basic timer (BT) for Oscillation stabilization control •

8bit Timers A/B

- One 8-bit timer/counter (**Timer A**) with three operating modes; Interval mode, capture mode and PWM mode.
- One 8-bit timer/counter (**Timer B**) Carrier frequency (or PWM) generator.

Watch Timer

- Real-time and interval time measurement.
- Four frequency output to BUZ pin.
- Clock generation for LCD.

LCD Controller/Driver (Optional)

- 8 COM × 19 SEG (MAX 19 digit)
  4 COM × 19 SEG (MAX 8 digit)

A/D Converter

- Nine analog input channels
- 12.5us conversion speed at 4MHz f_{ADC} clock.

Asynchronous UART

- Programmable baud rate generator
- Support serial data transmit/receive operations with 8-bit, 9-bit UART

Watchdog Timer

- Two oscillation sources selection (by Smart option)
- Safety work for noise interference

Low Voltage Reset (LVR)

- Low Voltage Check to make system reset
- V_{LVR} = 2.6V/3.3V/3.9V

Voltage Detector for Indication

- Voltage Detector to indicate specific voltage.
- S/W control (2.4V, 2.7V, 3.3V, 3.9V)

Operating Temperature Range

- –25°C to + 85°C

Operating Voltage Range

- 2.2V to 5.5 V at 4 MHz f_{OSC}
- 2.7V to 5.5 V at 8 MHz f_{OSC}

Package Type

- 32-pin SDIP, 32-pin SOP
- 42-pin SDIP, 44-pin QFP

Smart Option

- Low Voltage Reset(LVR) level and enable/disable are at your hardwired option.
- I/O Port (P0.0- P0.2/P3.3-P3.6) mode selection at Reset.
- Watchdog Timer oscillator selection.

BLOCK DIAGRAM

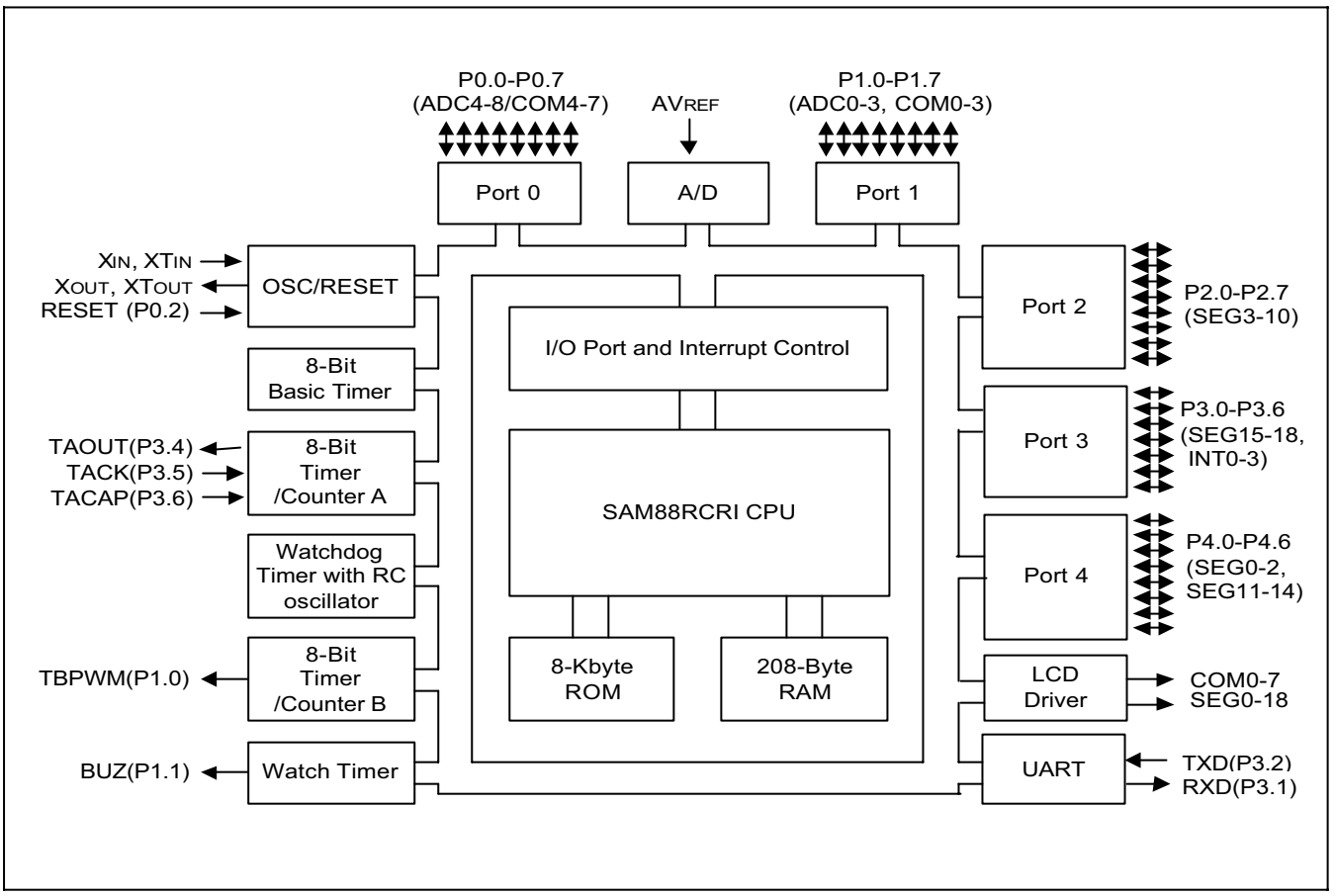


Figure 1-1. S3C9484/C9488/F9488 Block Diagram

PIN ASSIGNMENT

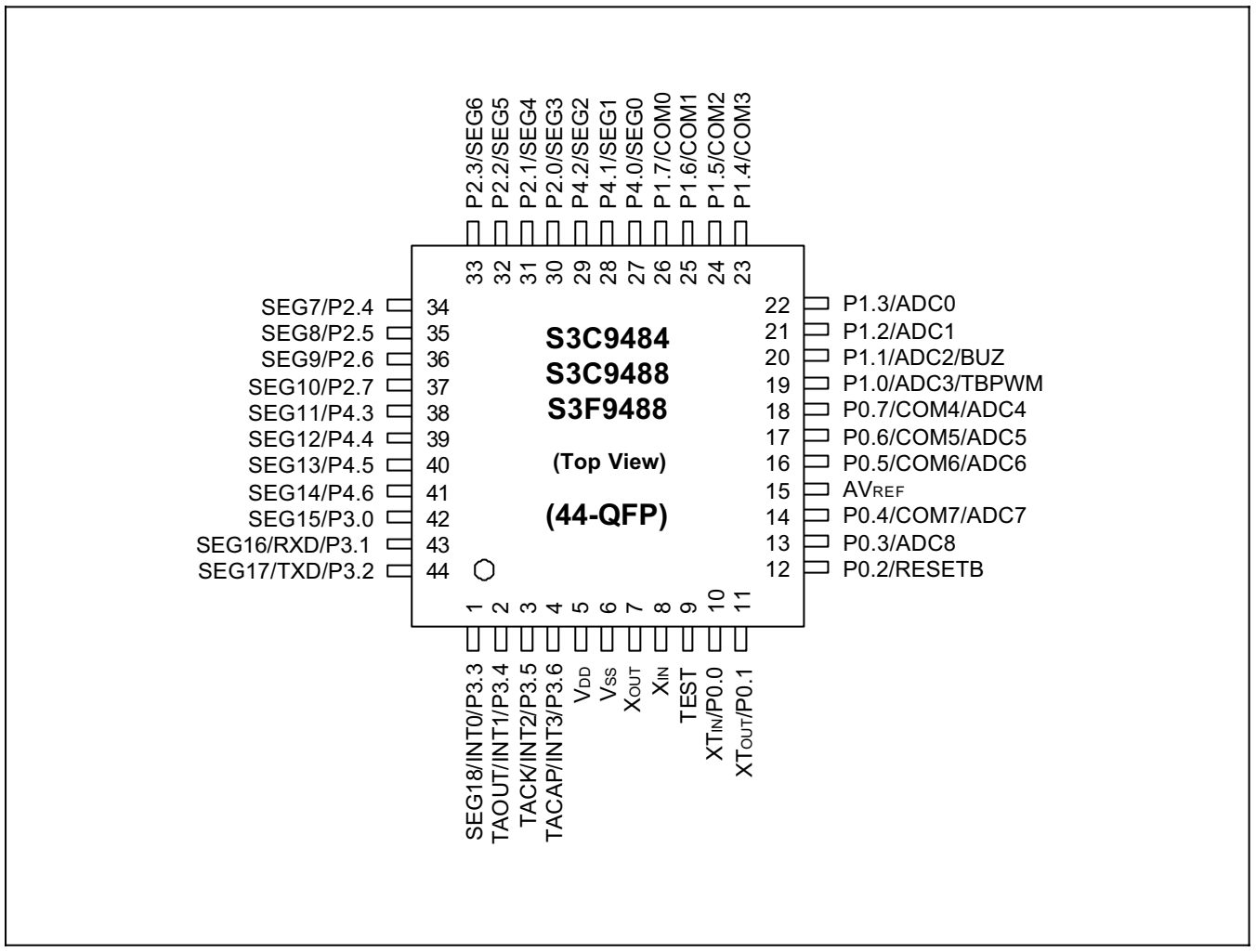


Figure 1-2. S3C9484/C9488/F9488 Pin Assignment (44-QFP)

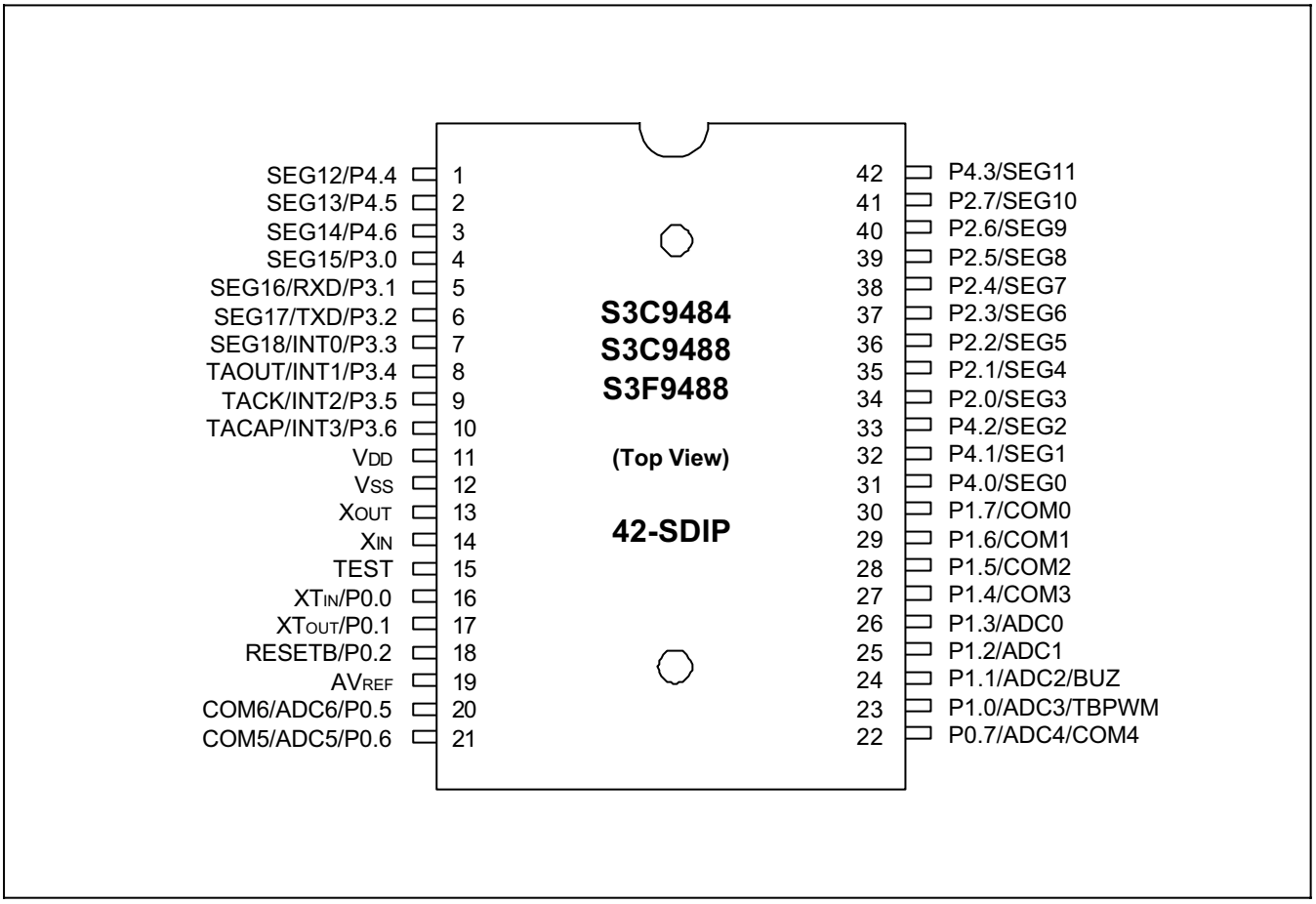


Figure 1-3. S3C9484/C9488/F9488 Pin Assignment (42-SDIP)

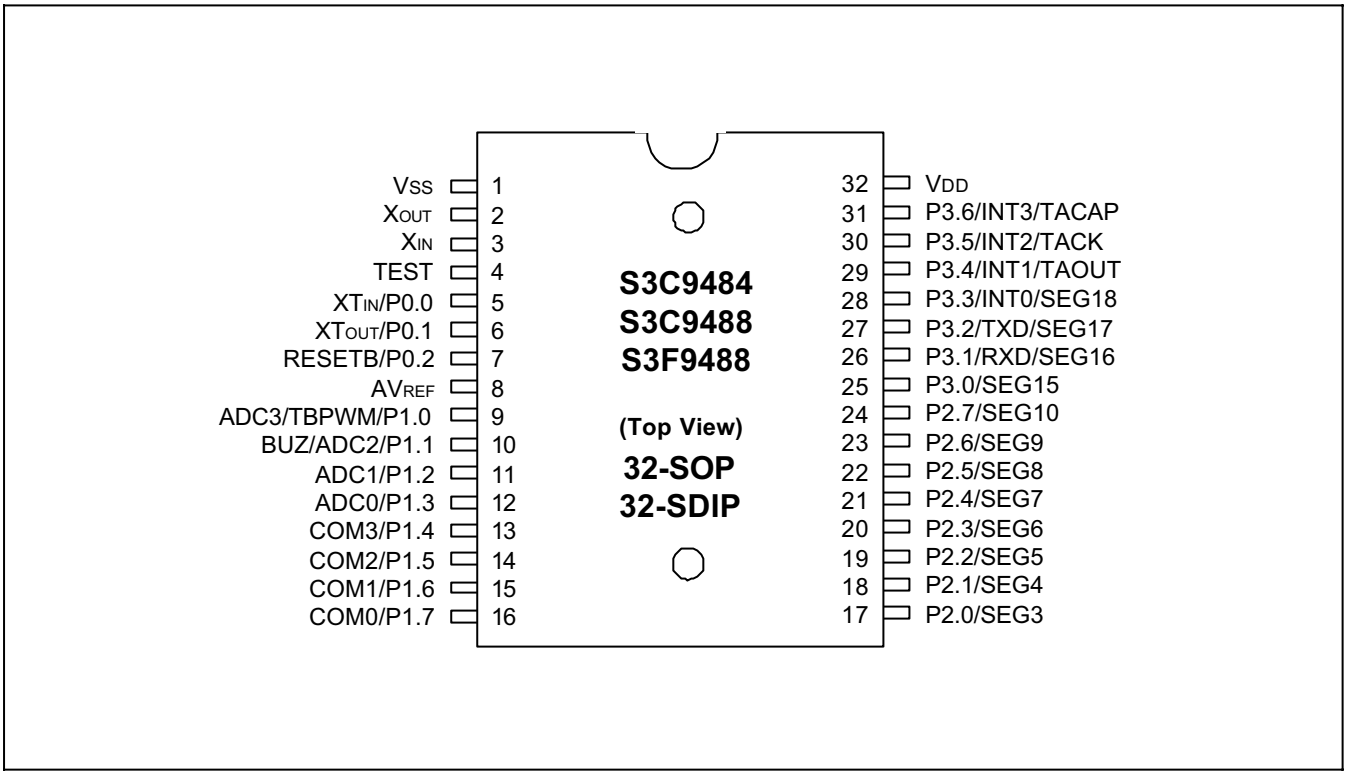


Figure 1-4. S3C9484/C9488/F9488 Pin Assignment (32-SOP/SDIP)

PIN DESCRIPTIONS

Table 1-1. Pin Descriptions of 44-QFP and 42-SDIP

Pin Names	Pin Type	Pin Description	Circuit Type	44 Pin No.	42 Pin No.	Shared Functions
P0.0, P0.1 P0.2 P0.3 P0.4 P0.5 P0.6 P0.7	I/O	I/O port with bit-programmable pins. Configurable to input or push-pull output mode. Pull-up resistors can be assigned by software. Pins can also be assigned individually as alternative function pins.	E E-1 E-2 H-16	10–14 16–18	16–18 20-22	XTIN, XTOUT RESETB ADC8 COM7/ADC7 COM6/ADC6 COM5/ADC5 COM4/ADC4
P1.0 P1.1–P1.3 P1.4–P1.7	I/O	I/O port with bit-programmable pins. Configurable to input or push-pull output mode. Pull-up resistors can be assigned by software. Pins can also be assigned individually as alternative function pins.	E-3 E-1 H-14	19–26	23–30	ADC3/TBPWM ADC2/BUZ ADC1–ADC0 COM3–COM0
P2.0–P2.7	I/O	I/O port with bit-programmable pins. Configurable to input mode, push-pull output mode. Input mode with pull-up resistors can be assigned by software. The port 2 pins have high current drive capability. Pins can also be assigned individually as alternative function pins.	H-14	30–37	34–41	SEG3–SEG10
P3.0–P3.2 P3.3 P3.4, P3.6 P3.5	I/O	I/O port with bit-programmable pins. Configurable to input or push-pull output mode. Pull-up resistors can be assigned by software. Pins can also be assigned individually as alternative function pins.	H-14 H-15 H-17 D-5 D-4	42–44, 1–4	4–10	SEG15 SEG16/RXD SEG17/TXD SEG18/INT0 TAOUT/INT1 TACK/INT2 TACAP/INT3
P4.0–P4.6	I/O	I/O port with bit-programmable pins. Configurable to input mode, push-pull output mode. Input mode with pull-up resistors can be assigned by software. Pins can also be assigned individually as alternative function pins.	H-14	27–29 38–41	31–33 42, 1–3	SEG0–2 SEG11–14
X_{IN}, X_{OUT}	I, O	System clock input and output pins	–	8,7	14,13	–
TEST	I	Test signal input pin (for factory use only; must be connected to V_{SS}.)	_	9	15	_
V_{DD}	–	Power supply input pin	–	5	11	–
V_{SS}	–	Ground pin	–	6	12	–

Table 1-1. Pin Descriptions of 44-QFP and 42-SDIP (Continued)

Pin Names	Pin Type	Pin Description	Circuit Type	44 Pin No.	42 Pin No.	Shared Functions
SEG0–18	O	LCD segment display signal output pins	H-14 H-15 H-17	27–44, 1	31–42, 1–7	P4.0–P4.2 P2.0–P2.7 P4.3–P4.6 P3.0 P3.1/RXD P3.2/TXD P3.3/INT0
COM0–7	O	LCD common signal output pins	H-14 H-16	26–23 18–16 14	30–27 20–22	P1.7–P1.4 P0.4–P0.7
ADC0–8	I	A/D converter analog input channels	E-1 E-3 H-16	22–20 19 18-14 13	20–26	P1.3–P1.2 P1.1/BUZ P1.0/TBPWM P0.7/COM4 P0.6/COM5 P0.5/COM6 P0.4/COM7 P0.3
AV_{REF}	I	A/D converter reference voltage		15	19	
RXD	I/O	Serial data RXD pin for receive input and transmit output (mode 0)	H-17	43	5	P3.1/SEG16
TXD	O	Serial data TXD pin for transmit output and shift clock output (mode 0)	H-17	44	6	P3.2/SEG17
INT0 INT1 INT2 INT3	I	External interrupts.	H-15 D-5 D-4	1–4	7–10	P3.3/SEG18 P3.4/TAOUT P3.5/TACK P3.6/TACAP
TAOUT	O	Timer/counter(A) overflow output, or Timer/counter(A) PWM output	D-5	2	8	P3.4/INT1
TACK	I	Timer/counter(A) external clock input	D-4	3	9	P3.5/INT2
TACAP	I	Timer/counter(A) external capture input	D-4	4	10	P3.6/INT3
BUZ	O	Frequency output to buzzer	E-3	20	24	P1.1/ADC2
TBPWM	O	Timer(B) PWM output	E-3	19	23	P1.0/ADC3
XT_{IN}, XT_{OUT}	I O	Clock input and output pins for subsystem clock	E	10 11	16 17	P0.0 P0.1
RESETB	I	System reset signal input pin	B	12	18	P0.2

Table 1-2. Pin Descriptions of 32-SOP and 32-SDIP

Pin Names	Pin Type	Pin Description	Circuit Type	32 Pin No.	Shared Functions
P0.0, P0.1 P0.2	I/O	I/O port with bit-programmable pins. Configurable to input or push-pull output mode. Pull-up resistors can be assigned by software. Pins can also be assigned individually as alternative function pins.	E E-2	5–7	XTIN, XTOUT RESETB
P1.0 P1.1–P1.3 P1.4–P1.7	I/O	I/O port with bit-programmable pins. Configurable to input or push-pull output mode. Pull-up resistors can be assigned by software. Pins can also be assigned individually as alternative function pins.	E-3 E-1 H-14	9–16	ADC3/TBPWM ADC2/BUZ ADC1–ADC0 COM3–COM0
P2.0–P2.7	I/O	I/O port with bit-programmable pins. Configurable to input mode, push-pull output mode, or n-channel open-drain output mode. Input mode with pull-up resistors can be assigned by software. The port 2 pins have high current drive capability. Pins can also be assigned individually as alternative function pins.	H-14	17–24	SEG3–SEG10
P3.0–P3.2 P3.3 P3.4 P3.5 P3.6	I/O	I/O port with bit-programmable pins. Configurable to input or push-pull output mode. Pull-up resistors can be assigned by software. Pins can also be assigned individually as alternative function pins.	H-14 H-15 H-17 D-5 D-4	25–31	SEG15 SEG16/RXD SEG17/TXD SEG18/INT0 TAOUT/INT1 TACK/INT2 TACAP/INT3
X_{IN}, X_{OUT}	I, O	System clock input and output pins	–	2,3	–
TEST	I	Test signal input pin (for factory use only; must be connected to V_{SS}.)	–	4	–
V_{DD}	–	Power supply input pin	–	32	–
V_{SS}	–	Ground pin	–	1	–

Table 1-2. Pin Descriptions of 32-SOP and 32-SDIP (Continued)

Pin Names	Pin Type	Pin Description	Circuit Type	32 Pin No.	Shared Functions
SEG3–10 SEG15–18	O	LCD segment display signal output pins	H-14 H-15 H-17	17–28	P2.0–P2.7 P3.0 P3.1/RXD P3.2/TXD P3.3/INT0
COM0–3	O	LCD common signal output pins	H-14	16–13	P1.7–P1.4
ADC0–3	I	A/D converter analog input channels	E-1 E-3	12–9	P1.3–P1.2 P1.1/BUZ P1.0/TBPWM
AV_{REF}	I	A/D converter reference voltage		8	
RXD	I/O	Serial data RXD pin for receive input and transmit output (mode 0)	H-17	26	P3.1/SEG16
TXD	O	Serial data TXD pin for transmit output and shift clock output (mode 0)	H-17	27	P3.2/SEG17
INT0 INT1 INT2 INT3	I	External interrupts.	H-15 D-5 D-4	28–31	P3.3/SEG18 P3.4/TAOUT P3.5/TACK P3.6/TACAP
TAOUT	O	Timer/counter(A) overflow output, or Timer/counter(A) PWM output	D-5	29	P3.4/INT1
TACK	I	Timer/counter(A) external clock input	D-4	30	P3.5/INT2
TACAP	I	Timer/counter(A) external capture input	D-4	31	P3.5/INT3
BUZ	O	Frequency output to buzzer	E-3	10	P1.1/ADC2
TBPWM	O	Timer(B) PWM output	E-3	9	P1.0/ADC3
XT_{IN}, XT_{OUT}	I O	Clock input and output pins for subsystem clock	E	5 6	P0.0 P0.1
RESETB	I	System reset signal input pin	B	7	P0.2

PIN CIRCUITS

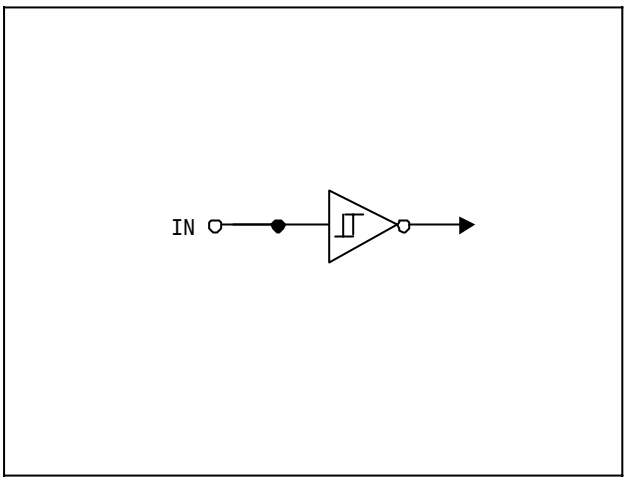


Figure 1-5. Pin Circuit Type B (RESET)

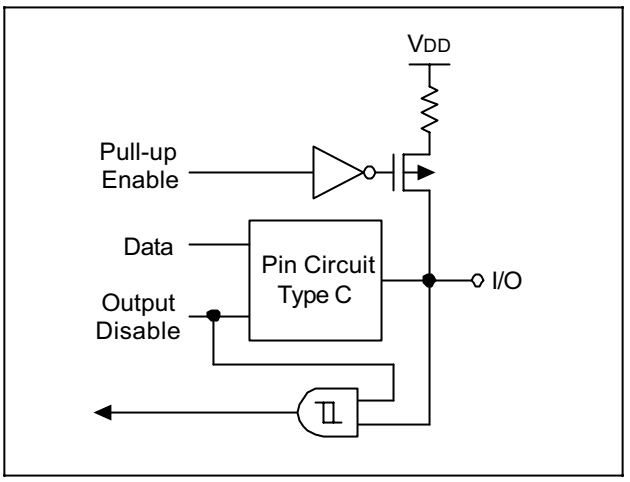


Figure 1-7. Pin Circuit Type D-2

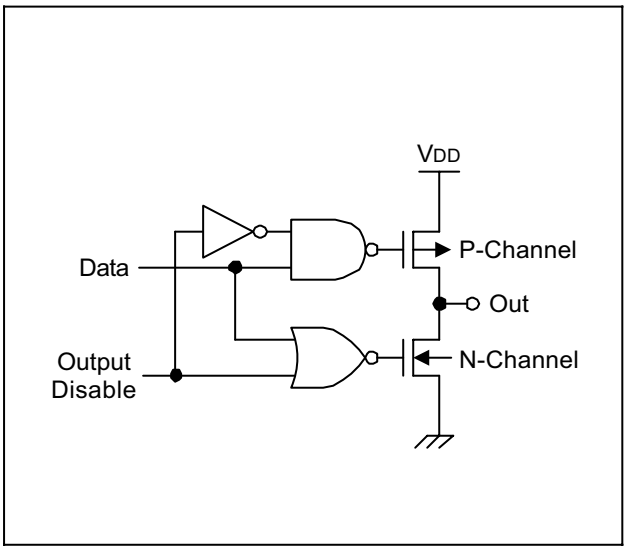


Figure 1-6. Pin Circuit Type C

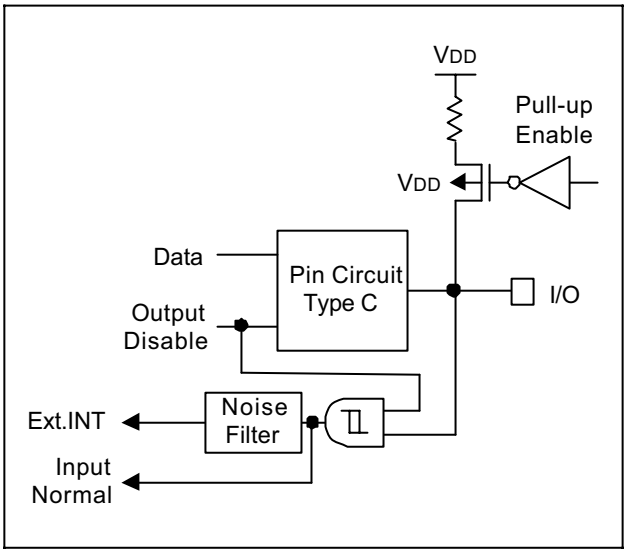


Figure 1-8. Pin Circuit Type D-4 (P3.5-P3.6)

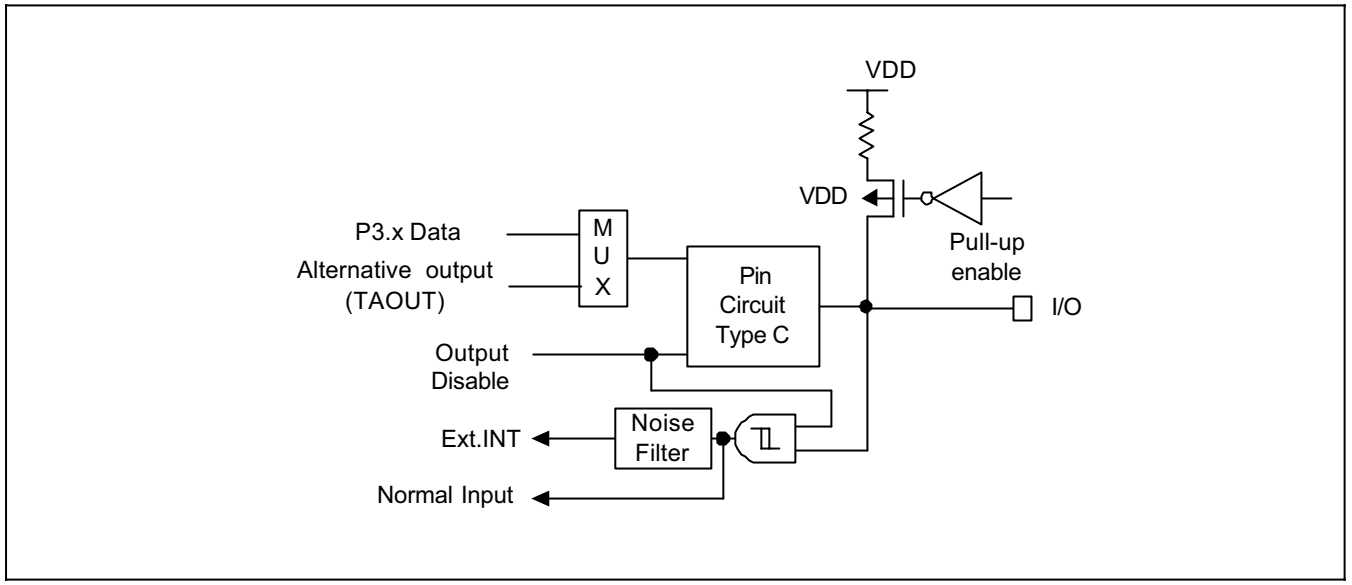


Figure 1-9. Pin Circuit Type D-5 (P3.4)

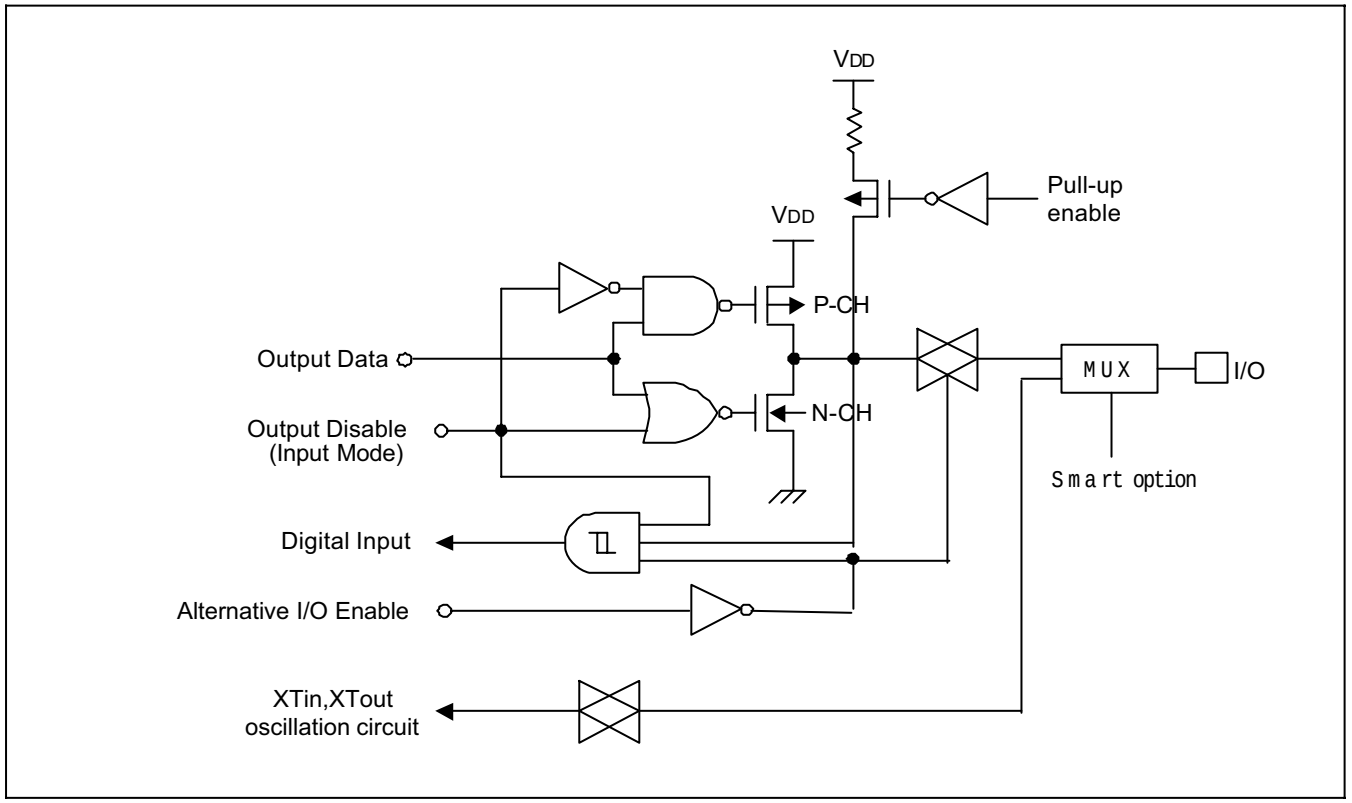


Figure 1-10. Pin Circuit Type E (P0.0, P0.1)

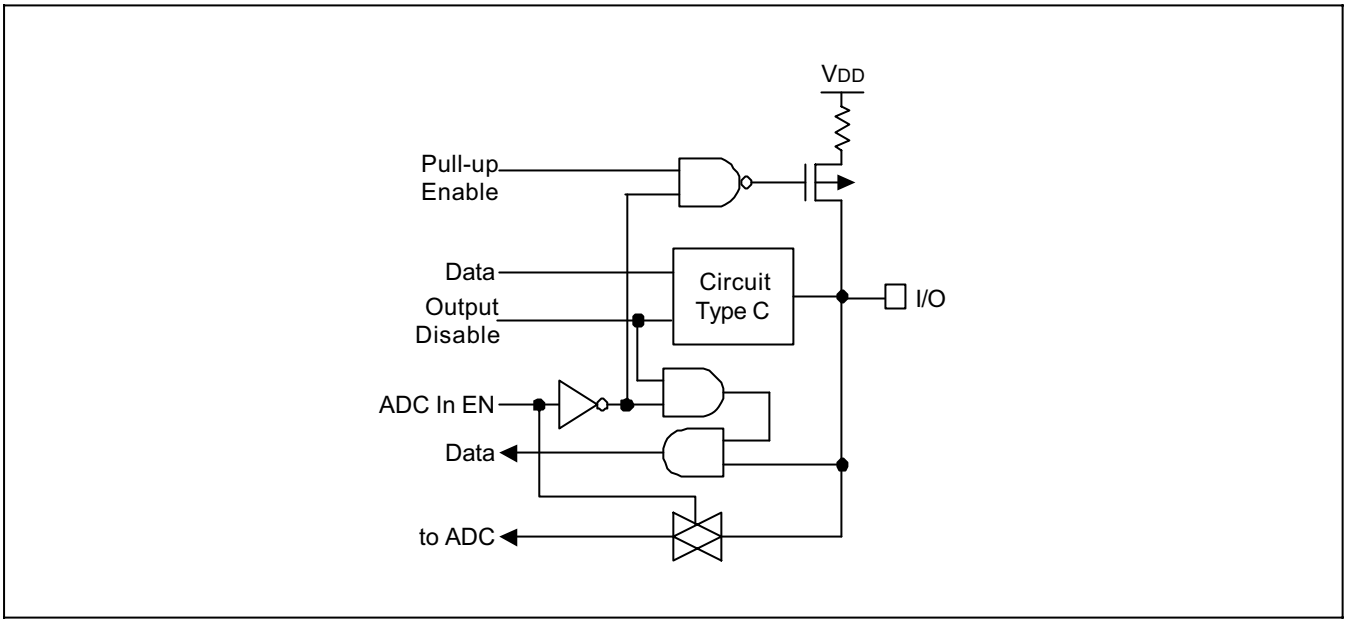


Figure 1-11. Pin Circuit Type E-1 (P0.3, P1.2–P1.3)

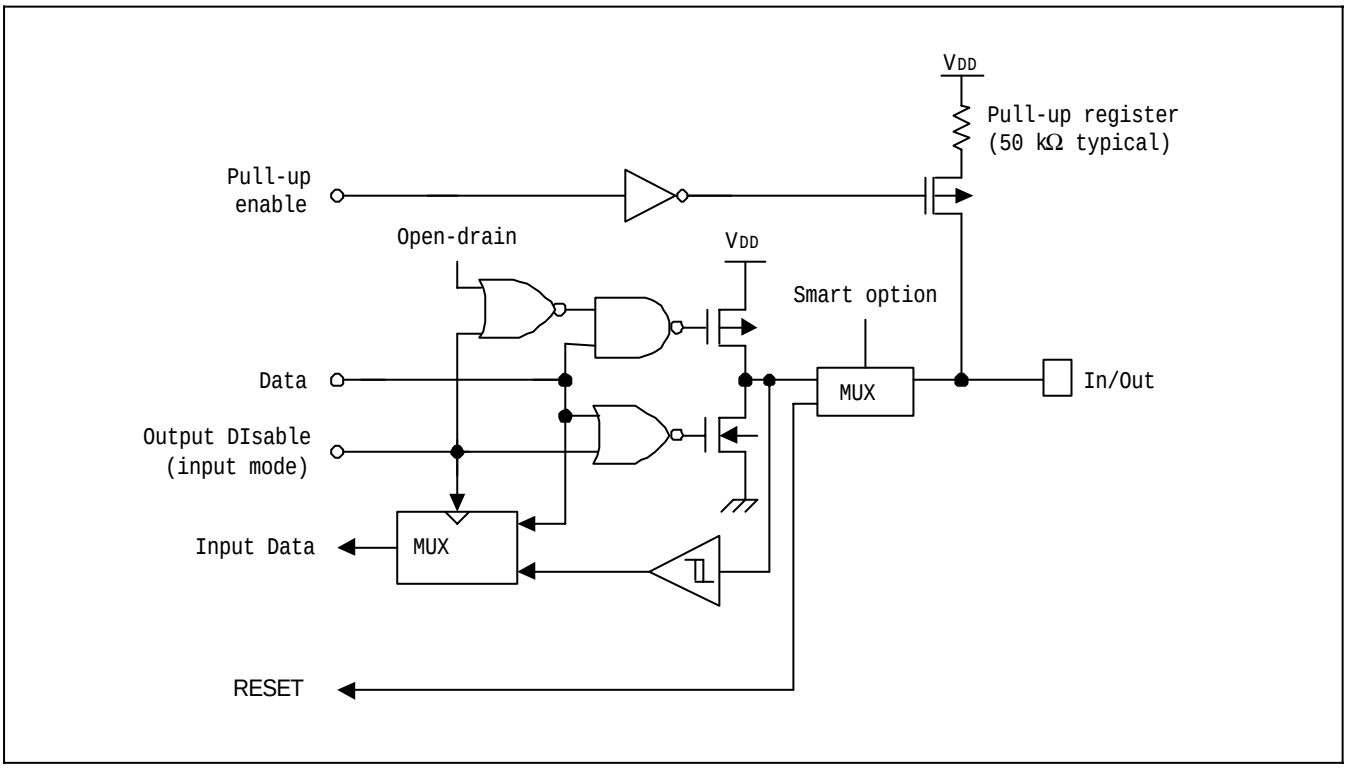


Figure 1-12. Pin Circuit Type E-2 (P0.2)

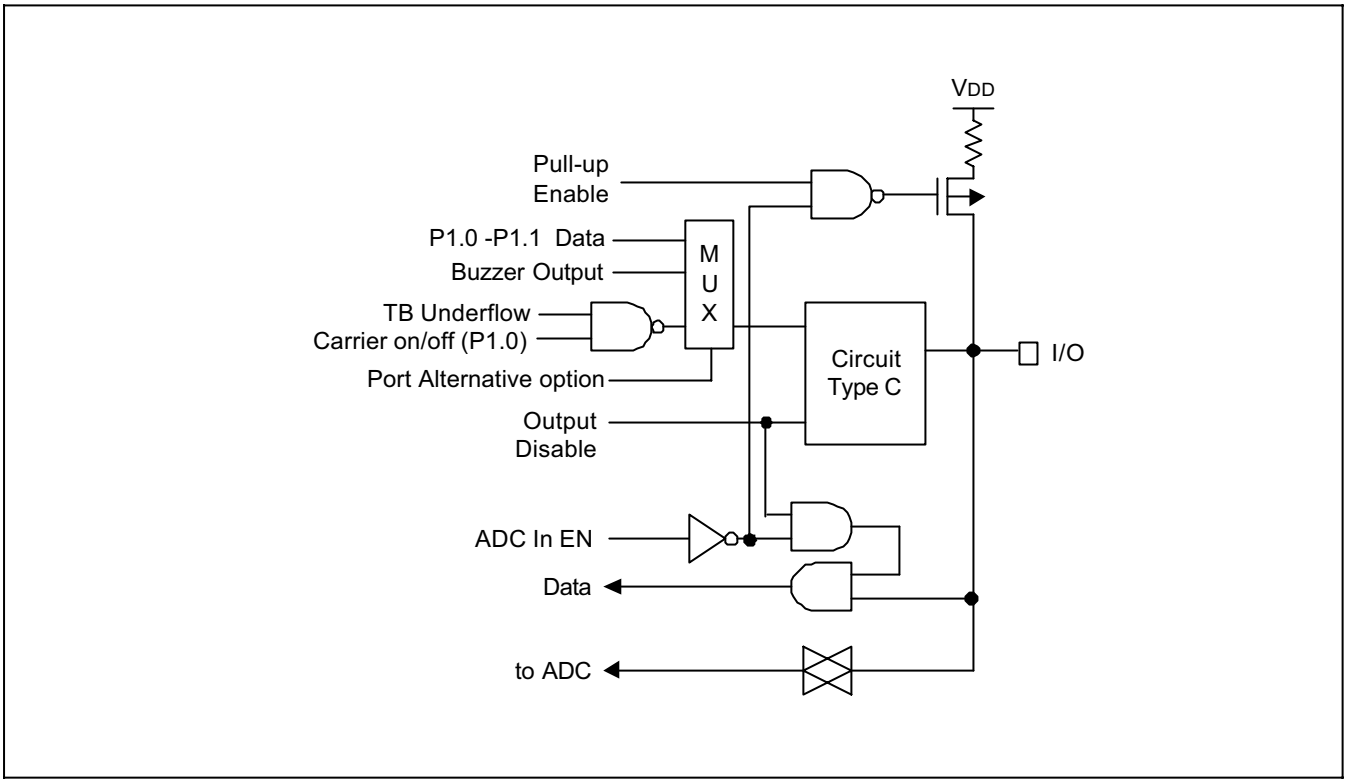


Figure 1-13. Pin Circuit Type E-3 (P1.0- P1.1)

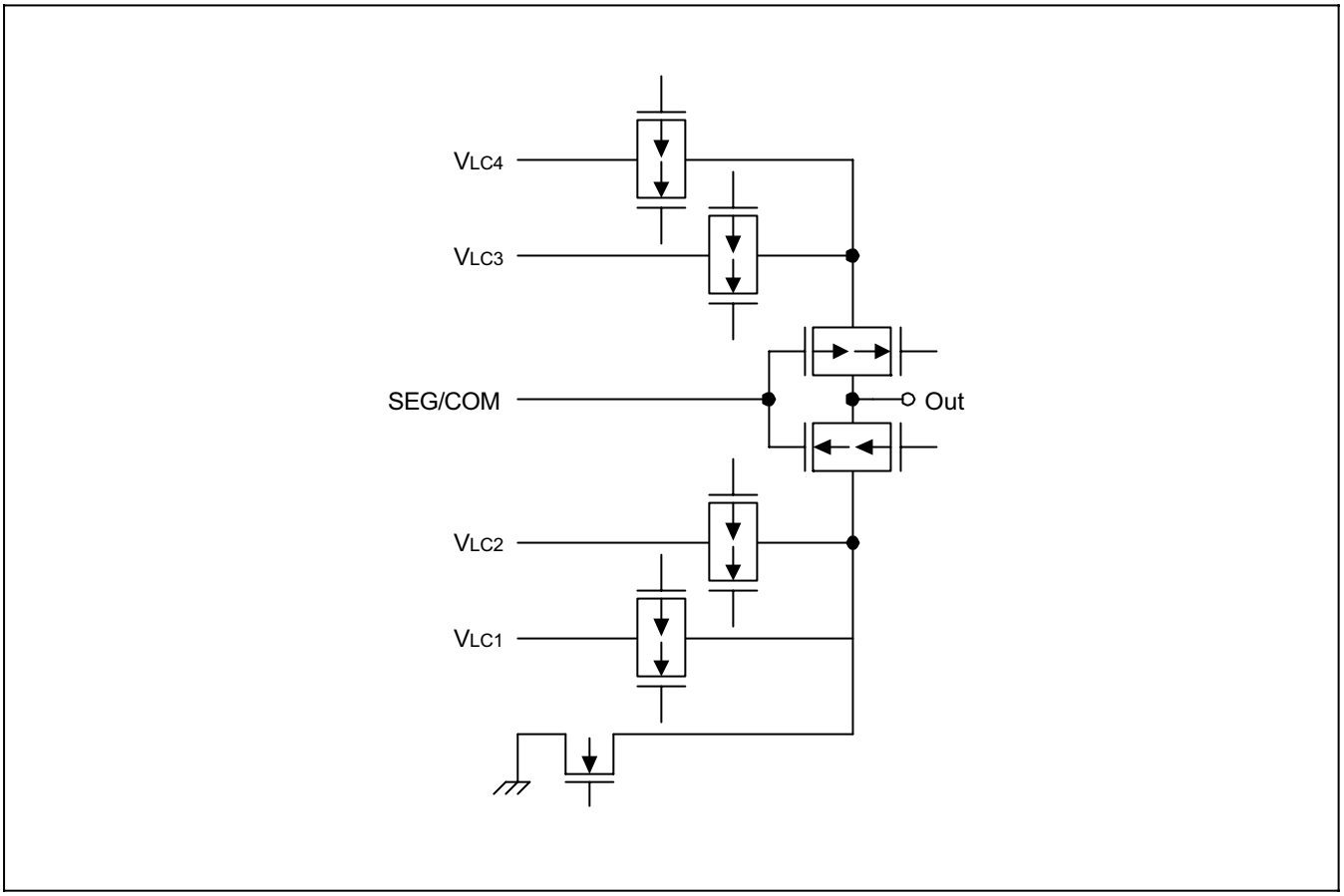


Figure 1-14. Pin Circuit Type H (SEG/COM)

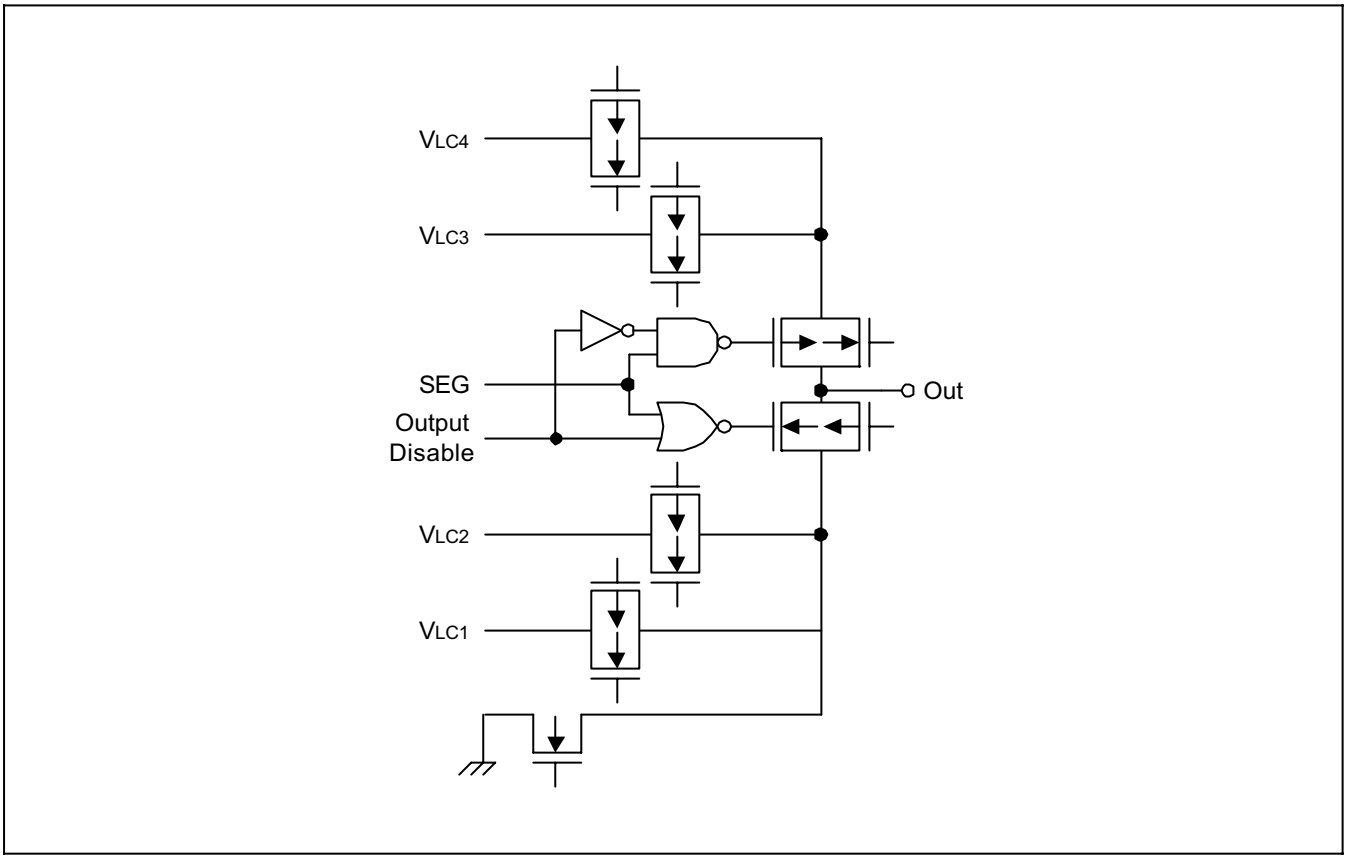


Figure 1-15. Pin Circuit Type H-4

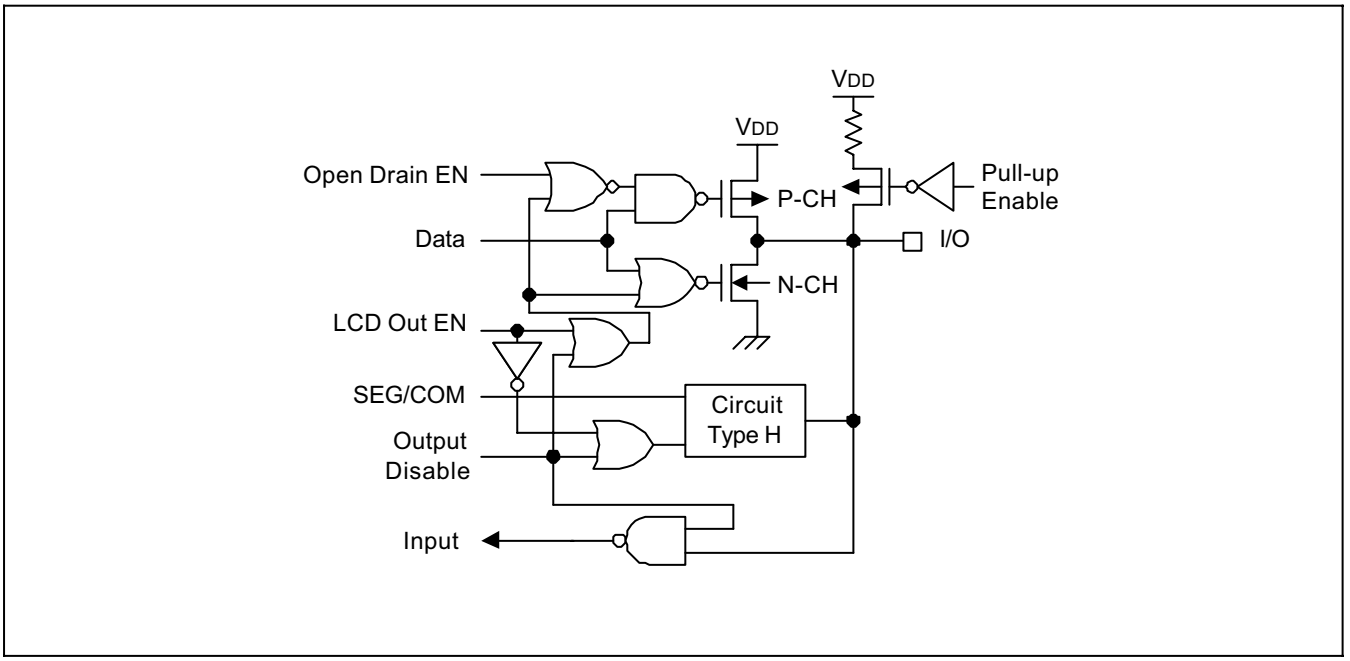


Figure 1-16. Pin Circuit Type H-14 (P1.4-P1.7, P2, P3.0, P4.0-P4.6)

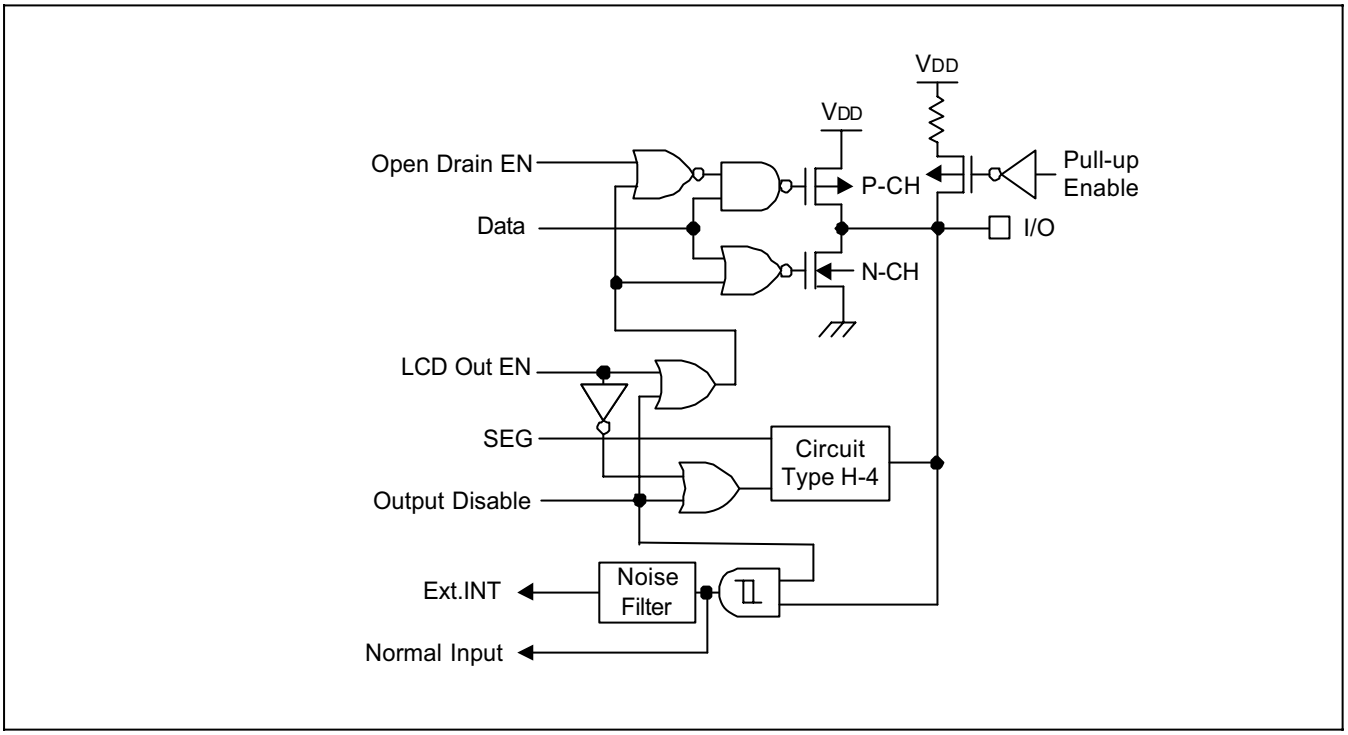


Figure 1-17. Pin Circuit Type H-15 (P3.3)

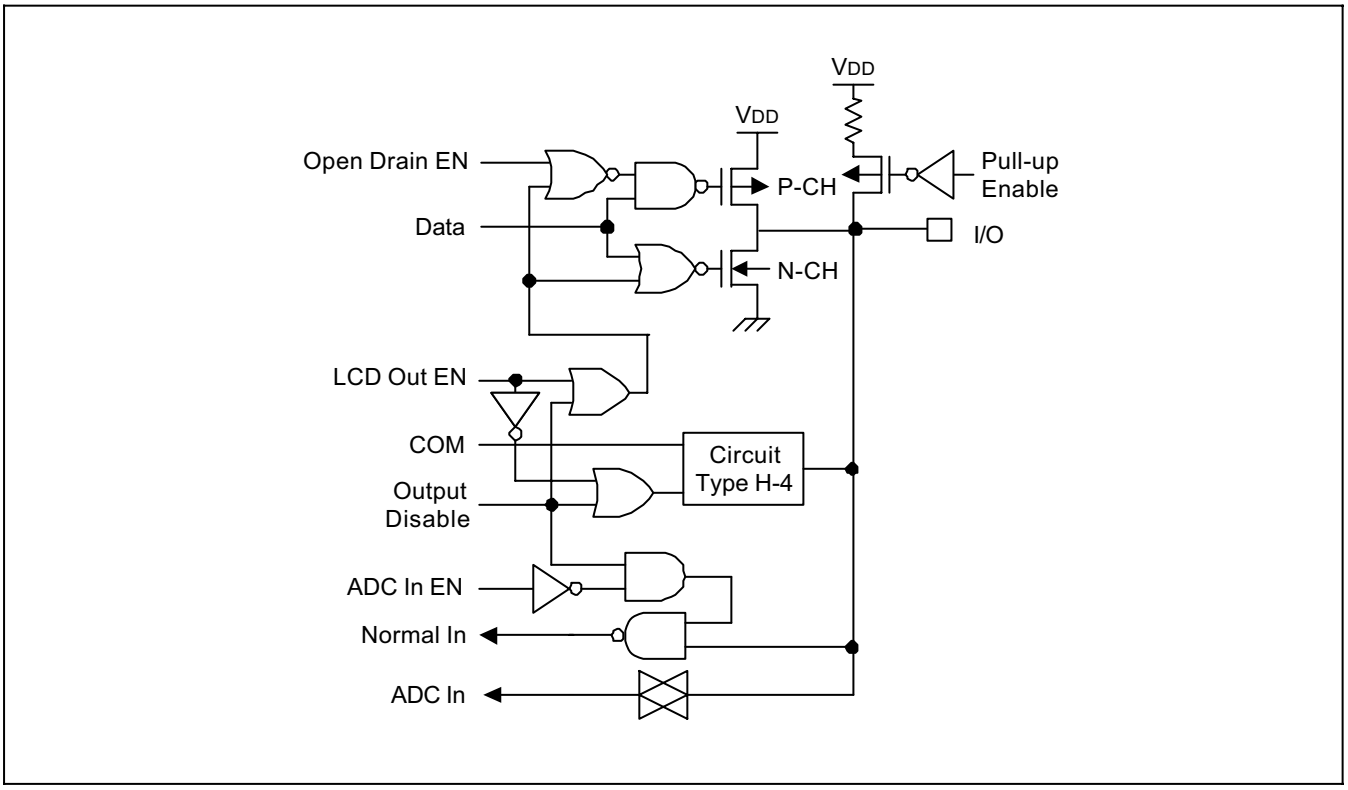


Figure 1-18. Pin Circuit Type H-16 (P0.4–P0.7)

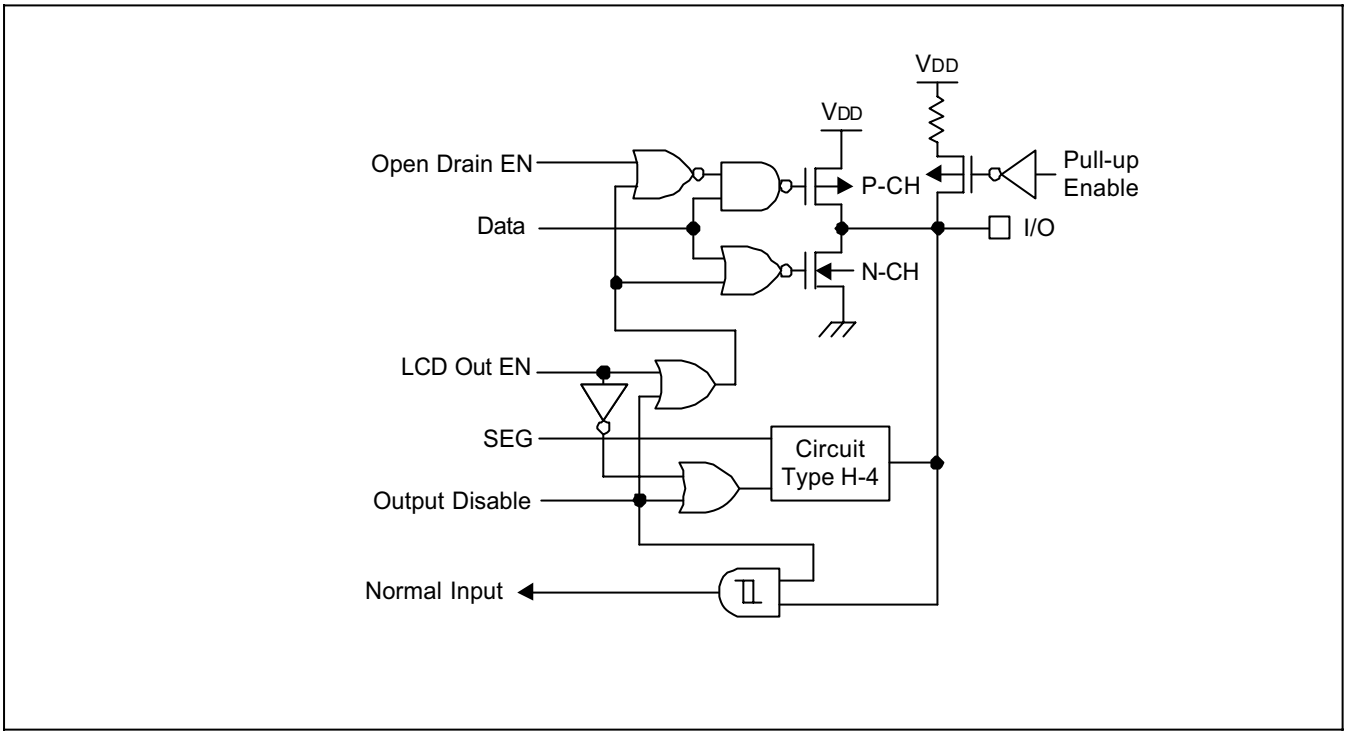


Figure 1-19. Pin Circuit Type H-17 (P3.1-P3.2)

2 ADDRESS SPACES

OVERVIEW

The S3C9484/C9488/F9488 microcontroller has two kinds of address space:

— Internal program memory (ROM)

— Internal register file

A 13-bit address bus supports program memory operations. A separate 8-bit register bus carries addresses and data between the CPU and the internal register file.

The S3F9488 have 8-Kbytes of on-chip program memory, which is configured as the Internal ROM mode, all of the 8-Kbyte internal program memory is used.

The S3C9484/C9488/F9488 microcontroller has 208 general-purpose registers in its internal register file. 47 bytes in the register file are mapped for system and peripheral control functions. And 19 bytes in the page1 is mapped for LCD display data area.

PROGRAM MEMORY (ROM)

Program memory (ROM) stores program codes or table data. The S3C9484/C9488 has 4K and 8Kbytes of internal mask programmable program memory. The program memory address range is therefore 0H–0FFFH and 0H-1FFFH. The S3F9488 have 8Kbytes (locations 0H–1FFFH) of internal multi time programmable (MTP) program memory (see Figure 2-1).

The first 2-bytes of the ROM (0000H–0001H) are interrupt vector address.
Unused locations (0002H–00FFH except 3CH, 3DH, 3EH, 3FH) can be used as normal program memory.
The location 3CH, 3DH, 3EH, and 3FH is used as smart option ROM cell.

The program reset address in the ROM is 0100H.

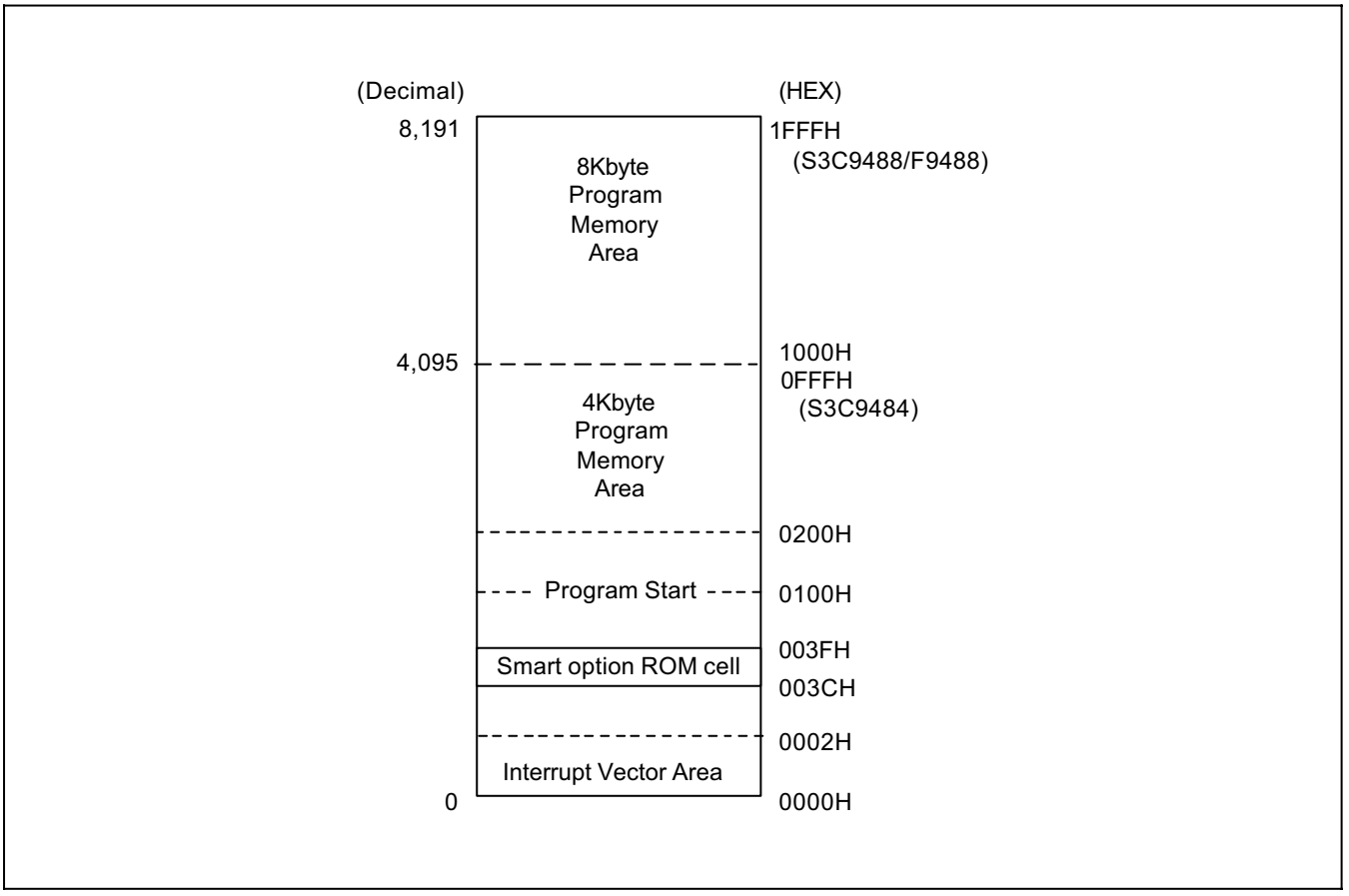
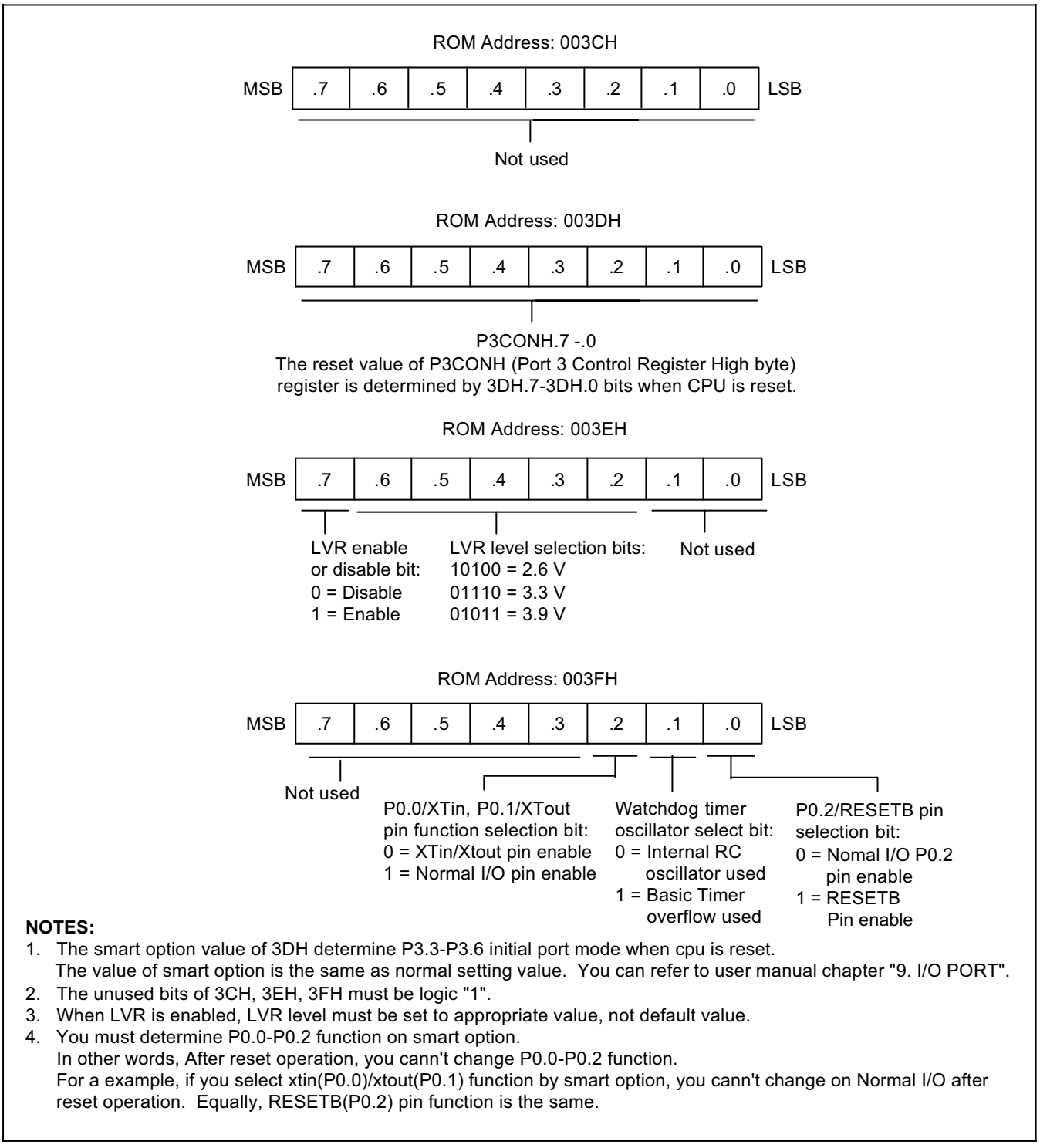


Figure 2-1. Program Memory Address Space

Smart Option

Smart option is the ROM option for starting condition of the chip. The ROM addresses used by smart option are from 003CH to 003FH. The default value of ROM is FFH.


ROM Address: 003CH
MSB .7 .6 .5 .4 .3 .2 .1 .0 LSB
Not used
ROM Address: 003DH
MSB .7 .6 .5 .4 .3 .2 .1 .0 LSB
P3CONH.7 -.0
The reset value of P3CONH (Port 3 Control Register High byte) register is determined by 3DH.7-3DH.0 bits when CPU is reset.
ROM Address: 003EH
MSB .7 .6 .5 .4 .3 .2 .1 .0 LSB
LVR enable or disable bit: 0 = Disable 1 = Enable
LVR level selection bits: 10100 = 2.6 V 01110 = 3.3 V 01011 = 3.9 V
Not used
ROM Address: 003FH
MSB .7 .6 .5 .4 .3 .2 .1 .0 LSB
Not used
P0.0/XTin, P0.1/XTout pin function selection bit: 0 = XTin/Xtout pin enable 1 = Normal I/O pin enable
Watchdog timer oscillator select bit: 0 = Internal RC oscillator used 1 = Basic Timer overflow used
P0.2/RESETB pin selection bit: 0 = Nomal I/O P0.2 pin enable 1 = RESETB Pin enable

NOTES:

1. The smart option value of 3DH determine P3.3-P3.6 initial port mode when cpu is reset.
   The value of smart option is the same as normal setting value. You can refer to user manual chapter "9. I/O PORT".
2. The unused bits of 3CH, 3EH, 3FH must be logic "1".
3. When LVR is enabled, LVR level must be set to appropriate value, not default value.
4. You must determine P0.0-P0.2 function on smart option.
   In other words, After reset operation, you cann't change P0.0-P0.2 function.
   For a example, if you select xtin(P0.0)/xtout(P0.1) function by smart option, you cann't change on Normal I/O after reset operation. Equally, RESETB(P0.2) pin function is the same.

Figure 2-2. Smart Option

REGISTER ARCHITECTURE

The upper 64-bytes of the S3C9484/C9488/F9488's internal register file are addressed as working registers, system control registers and peripheral control registers. The lower 192-bytes of internal register file (00H–BFH) is called the *general-purpose register space*. 274 registers in this space can be accessed; 208 are available for general-purpose use. And 19 are available for LCD display register. But if LCD driver not used, available for general-purpose use.

For many SAM88RCRI microcontrollers, the addressable area of the internal register file is further expanded by additional register pages at space of the general purpose register (00H–BFH). This register file expansion is not implemented in the S3C9484/C9488/F9488, however.

The specific register types and the area (in bytes) that they occupy in the internal register file are summarized in Table 2-1.

Table 2-1. Register Type Summary

Register Type	Number of Bytes
System and peripheral registers (page0 & page1)	47
General-purpose registers (including the 16-bit common working register area)	208
LCD display Registers (page1)	19
Total Addressable Bytes	**274**

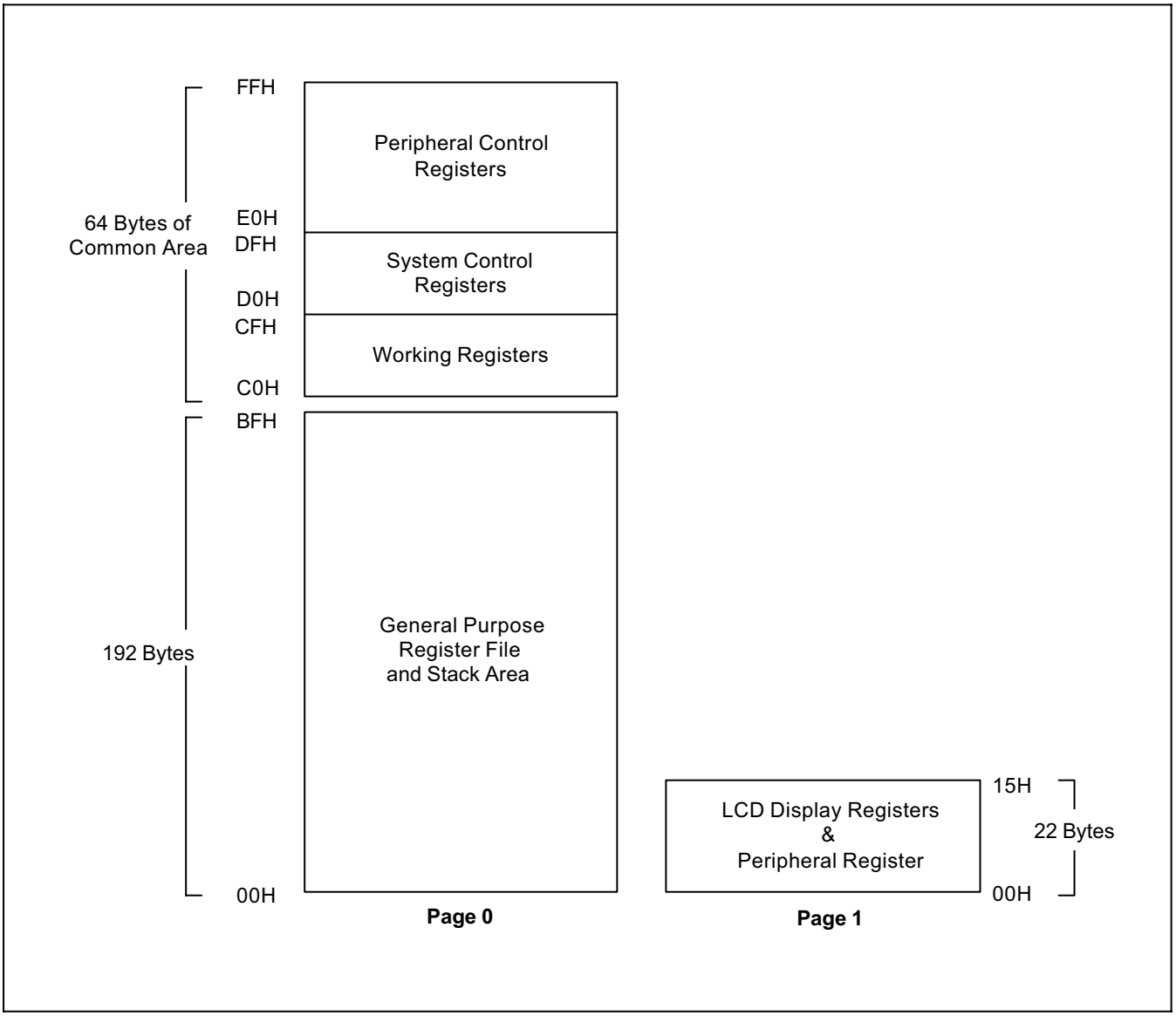


Figure 2-3. Internal Register File Organization

COMMON WORKING REGISTER AREA (C0H–CFH)

The SAM88RCRI register architecture provides an efficient method of working register addressing that takes full advantage of shorter instruction formats to reduce execution time.

This 16-byte address range is called common area. That is, locations in this area can be used as working registers by operations that address any location on any page in the register file.

Registers are addressed either as a single 8-bit register or as a paired 16-bit register. In 16-bit register pairs, the address of the first 8-bit register is always an even number and the address of the next register is an odd number. The most significant byte of the 16-bit data is always stored in the even-numbered register; the least significant byte is always stored in the next (+ 1) odd-numbered register.

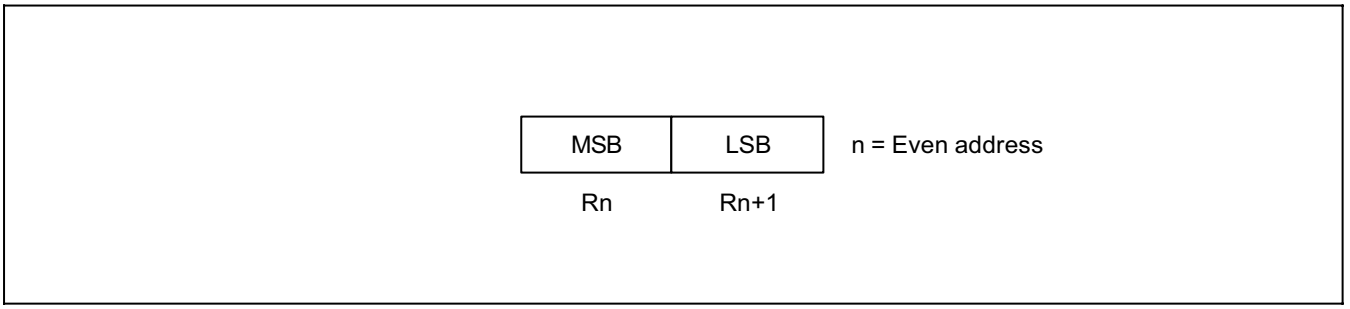


Figure 2-4. 16-Bit Register Pairs

SYSTEM STACK

S3F9-series microcontrollers use the system stack for subroutine calls and returns and to store data. The PUSH and POP instructions are used to control system stack operations. The S3C9484/C9488/F9488 architecture supports stack operations in the internal register file.

Stack Operations

Return addresses for procedure calls and interrupts and data are stored on the stack. The contents of the PC are saved to stack by a CALL instruction and restored by the RET instruction. When an interrupt occurs, the contents of the PC and the FLAGS registers are pushed to the stack. The IRET instruction then pops these values back to their original locations. The stack address always decrements *before* a push operation and increments *after* a pop operation. The stack pointer (SP) always points to the stack frame stored on the top of the stack, as shown in Figure 2-5.

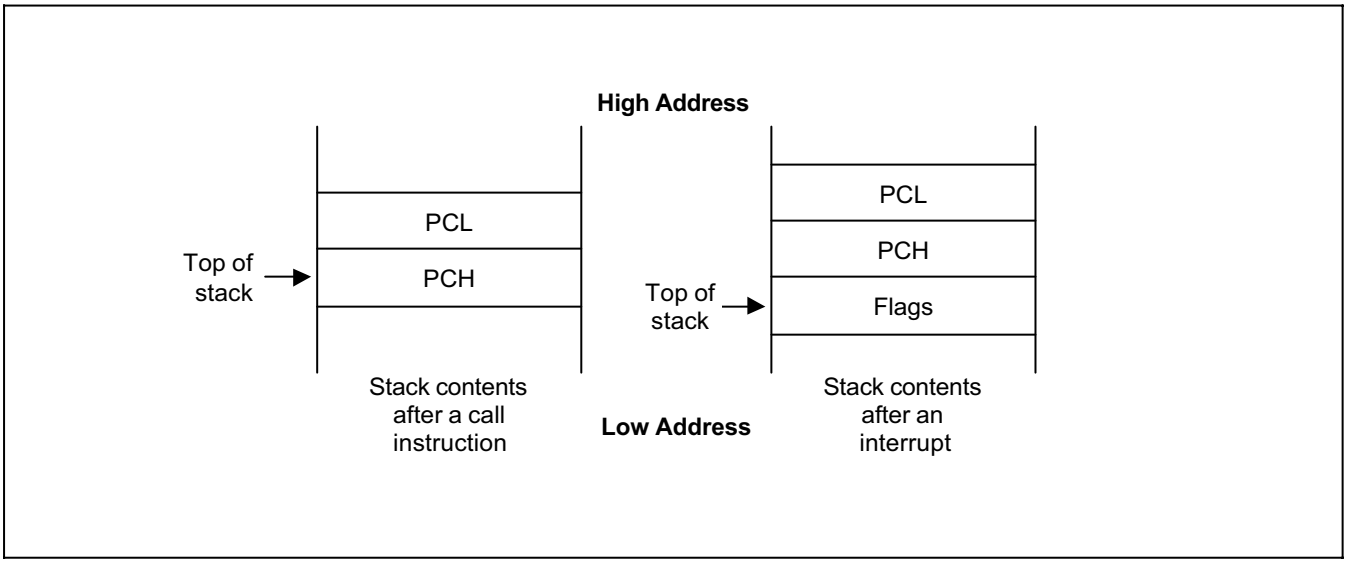


Figure 2-5. Stack Operations

Stack Pointer (SP)

Register location D9H contains the 8-bit stack pointer (SP) that is used for system stack operations. After a reset, the SP value is undetermined.

Because only internal memory space is implemented in the S3C9484/C9488/F9488, the SP must be initialized to an 8-bit value in the range 00H–0C0H.

NOTE

In case a Stack Pointer is initialized to 00H, it is decreased to FFH when stack operation starts. This means that a Stack Pointer access invalid stack area. We recommend that a stack pointer is initialized to C0H to set upper address of stack to BFH.

+ PROGRAMMING TIP — Standard Stack Operations Using PUSH and POP

The following example shows you how to perform stack operations in the internal register file using PUSH and POP instructions:

```
LD      SP,#0C0H        ;  SP ← C0H (Normally, the SP is set to C0H by the
                        ;           initialization routine)
•
•
•
PUSH    SYM             ;  Stack address 0BFH ← SYM
PUSH    R15             ;  Stack address 0BEH ← R15
PUSH    20H             ;  Stack address 0BDH ← 20H
PUSH    R3              ;  Stack address 0BCH ← R3
•
•
•
POP     R3              ;  R3 ← Stack address 0BCH
POP     20H             ;  20H ← Stack address 0BDH
POP     R15             ;  R15 ← Stack address 0BEH
POP     SYM             ;  SYM ← Stack address 0BFH
```

3 ADDRESSING MODES

OVERVIEW

Instructions that are stored in program memory are fetched for execution using the program counter. Instructions indicate the operation to be performed and the data to be operated on. Addressing mode is the method used to determine the location of the data operand. The operands specified in SAM88RC instructions may be condition codes, immediate data, or a location in the register file, program memory, or data memory.

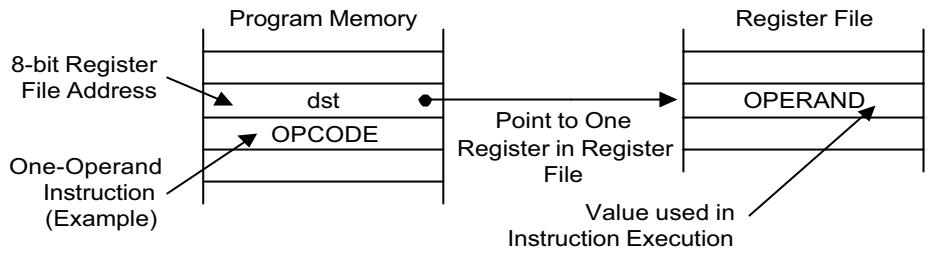
The S3C-series instruction set supports seven explicit addressing modes. Not all of these addressing modes are available for each instruction. The seven addressing modes and their symbols are:

— Register (R)
— Indirect Register (IR)
— Indexed (X)
— Direct Address (DA)
— Indirect Address (IA)
— Relative Address (RA)
— Immediate (IM)

REGISTER ADDRESSING MODE (R)

In Register addressing mode (R), the operand value is the content of a specified register or register pair (see Figure 3-1).

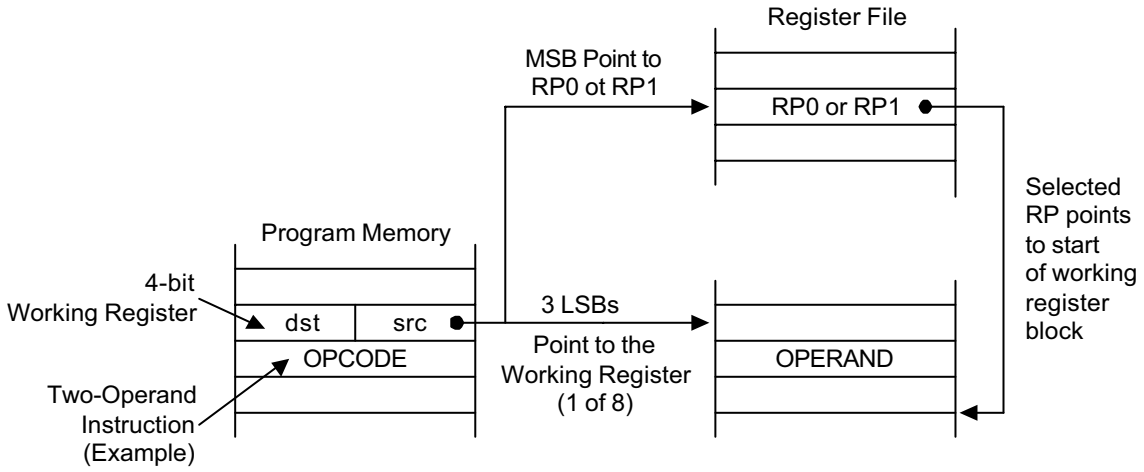
Working register addressing differs from Register addressing in that it uses a register pointer to specify an 8-byte working register space in the register file and an 8-bit register within that space (see Figure 3-2).



Sample Instruction:

```
DEC     CNTR            ;  Where CNTR is the label of an 8-bit register address
```

Figure 3-1. Register Addressing



Sample Instruction:

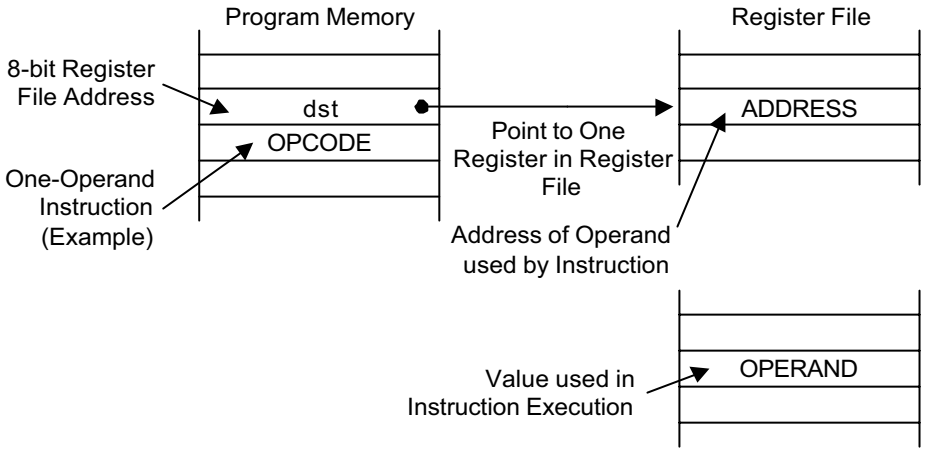
```
ADD     R1, R2          ;  Where R1 and R2 are registers in the currently
                           selected working register area.
```

Figure 3-2. Working Register Addressing

INDIRECT REGISTER ADDRESSING MODE (IR)

In Indirect Register (IR) addressing mode, the content of the specified register or register pair is the address of the operand. Depending on the instruction used, the actual address may point to a register in the register file, to program memory (ROM), or to an external memory space (see Figures 3-3 through 3-6).

You can use any 8-bit register to indirectly address another register. Any 16-bit register pair can be used to indirectly address another memory location.



Sample Instruction:

```
RL      @SHIFT          ;  Where SHIFT is the label of an 8-bit register address
```

Figure 3-3. Indirect Register Addressing to Register File

INDIRECT REGISTER ADDRESSING MODE (Continued)

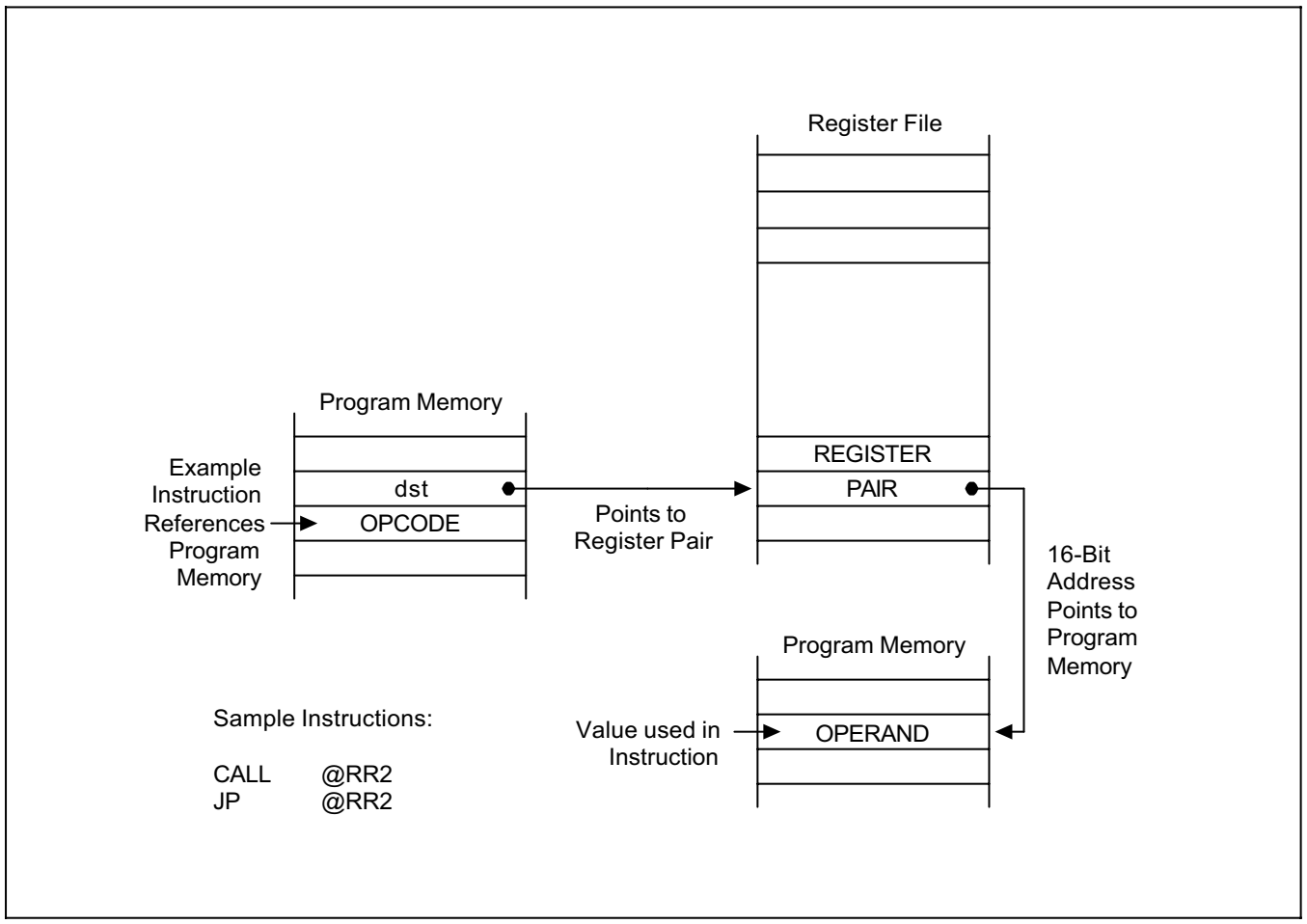


Figure 3-4. Indirect Register Addressing to Program Memory

INDIRECT REGISTER ADDRESSING MODE (Continued)

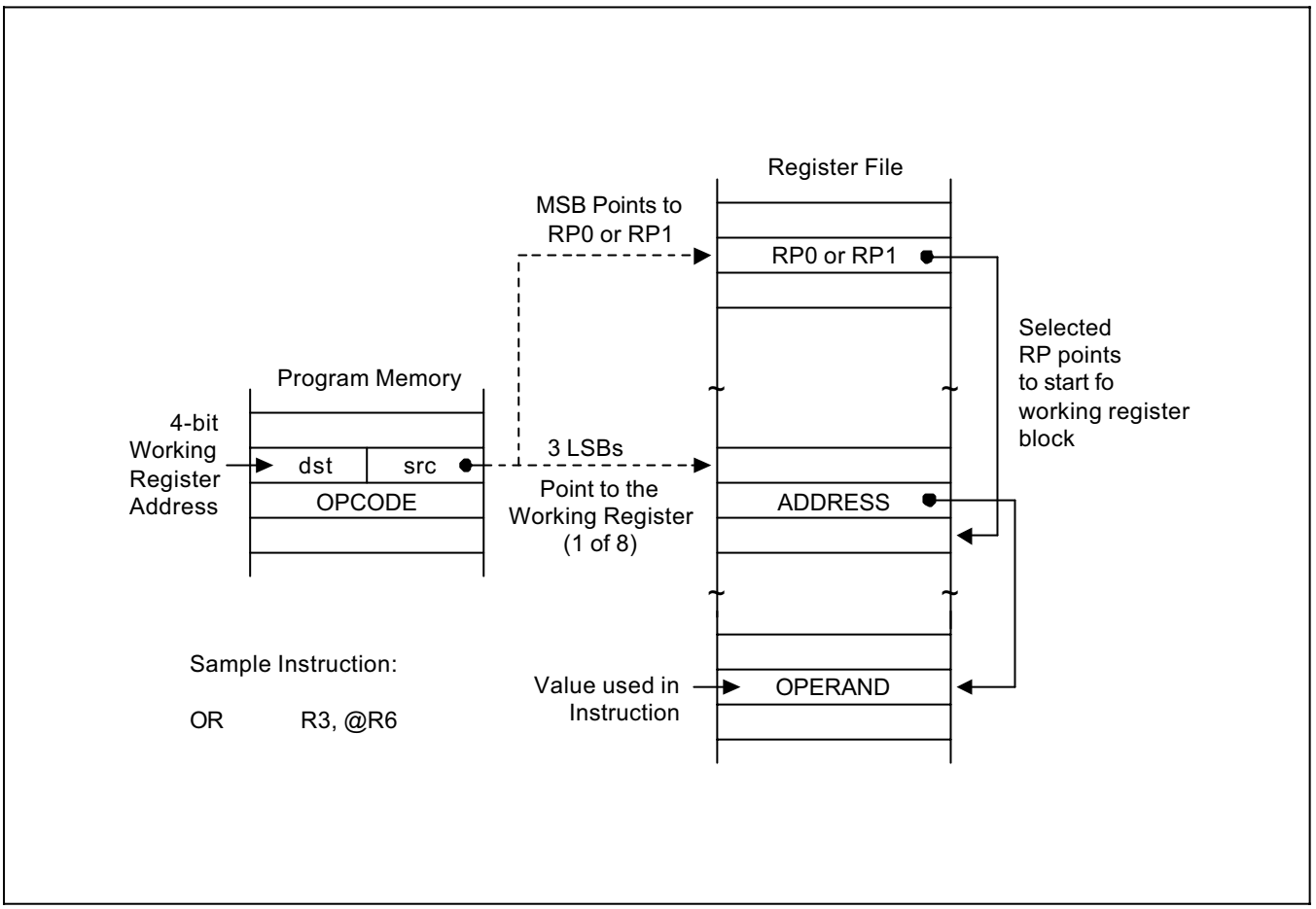


Figure 3-5. Indirect Working Register Addressing to Register File

INDIRECT REGISTER ADDRESSING MODE (Concluded)

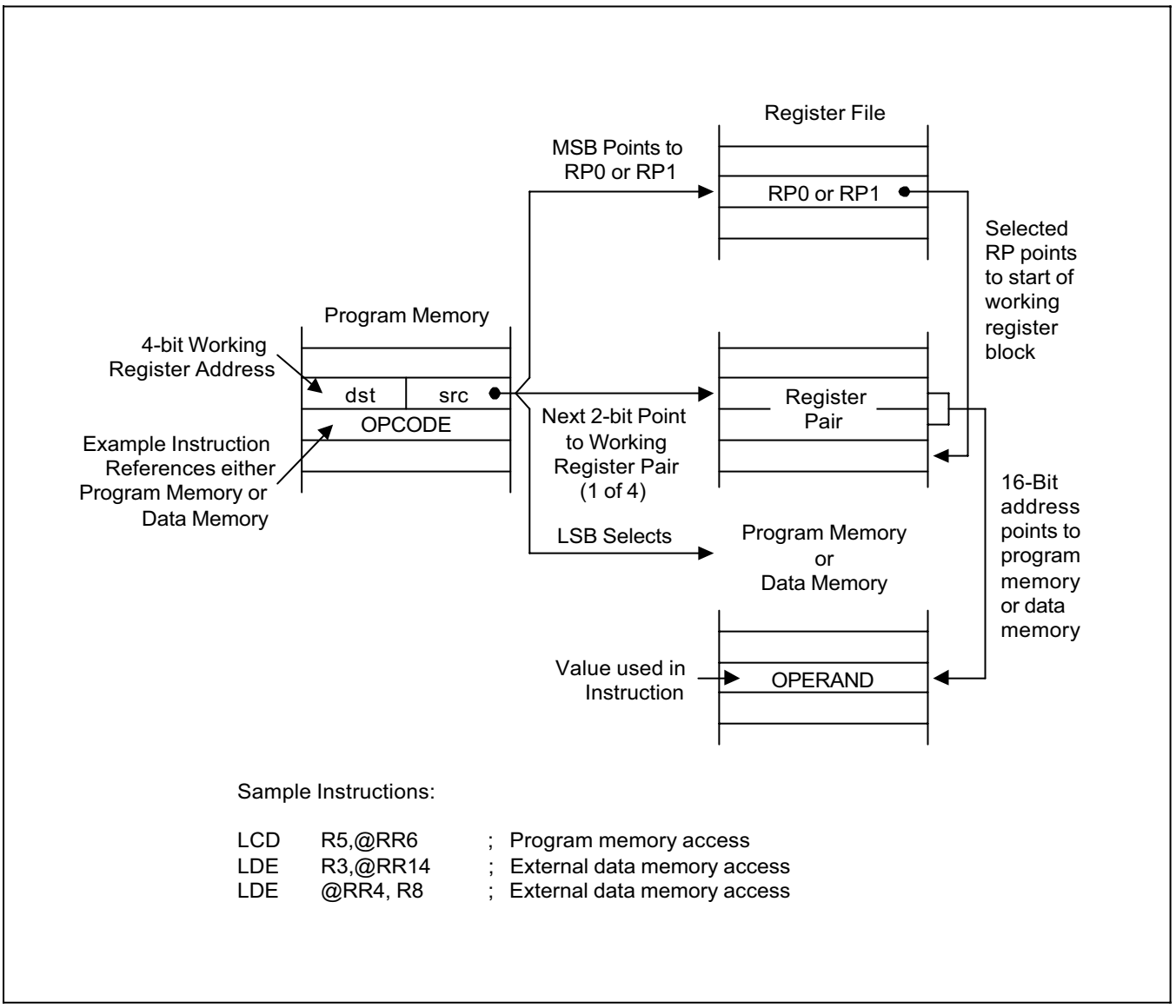


Figure 3-6. Indirect Working Register Addressing to Program or Data Memory

INDEXED ADDRESSING MODE (X)

Indexed (X) addressing mode adds an offset value to a base address during instruction execution in order to calculate the effective operand address (see Figure 3-7). You can use Indexed addressing mode to access locations in the internal register file or in external memory.

In short offset Indexed addressing mode, the 8-bit displacement is treated as a signed integer in the range –128 to +127. This applies to external memory accesses only (see Figure 3-8.)

For register file addressing, an 8-bit base address provided by the instruction is added to an 8-bit offset contained in a working register. For external memory accesses, the base address is stored in the working register pair designated in the instruction. The 8-bit or 16-bit offset given in the instruction is then added to that base address (see Figure 3-9).

The only instruction that supports Indexed addressing mode for the internal register file is the Load instruction (LD). The LDC and LDE instructions support Indexed addressing mode for internal program memory and for external data memory, when implemented.

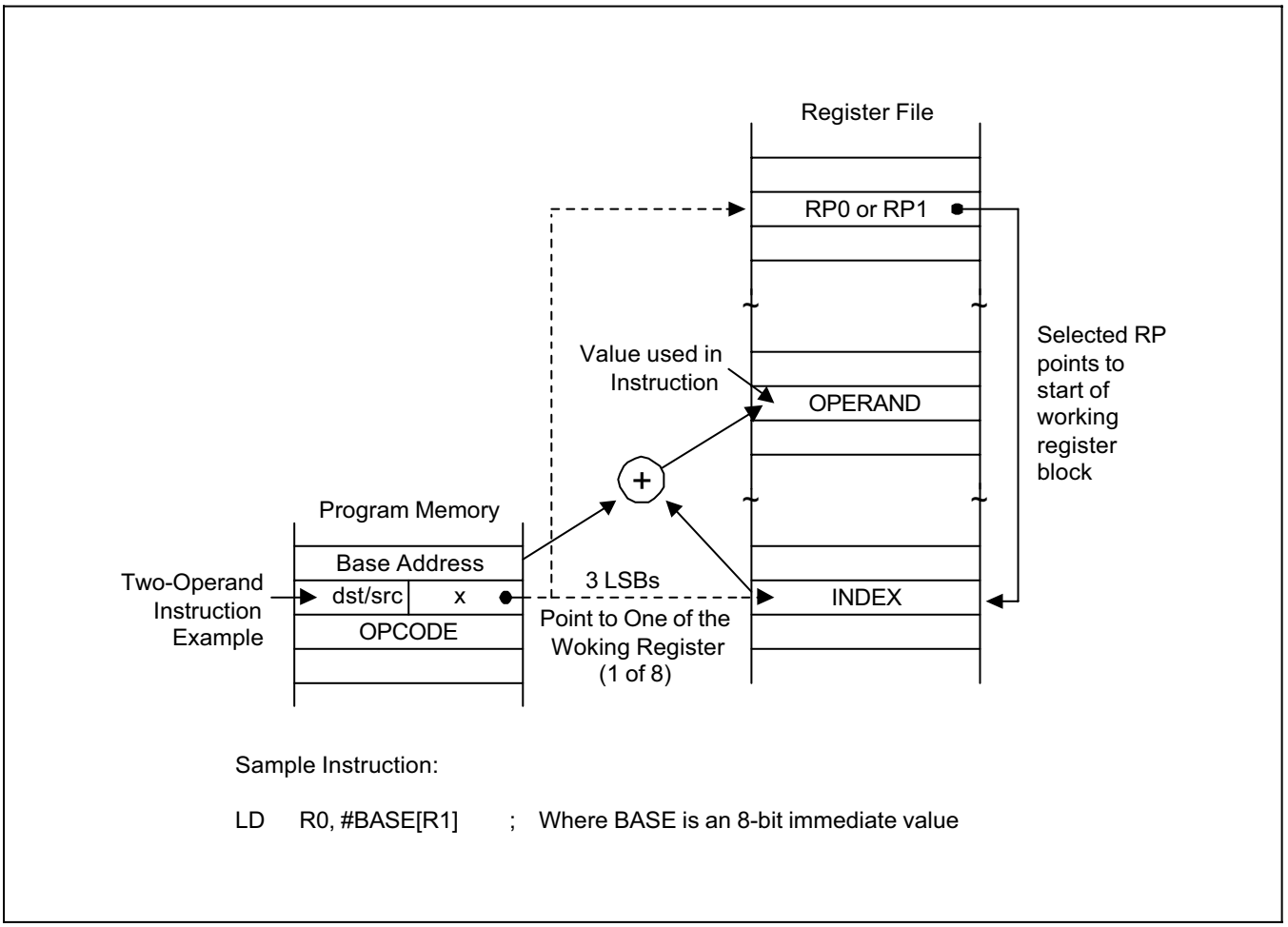
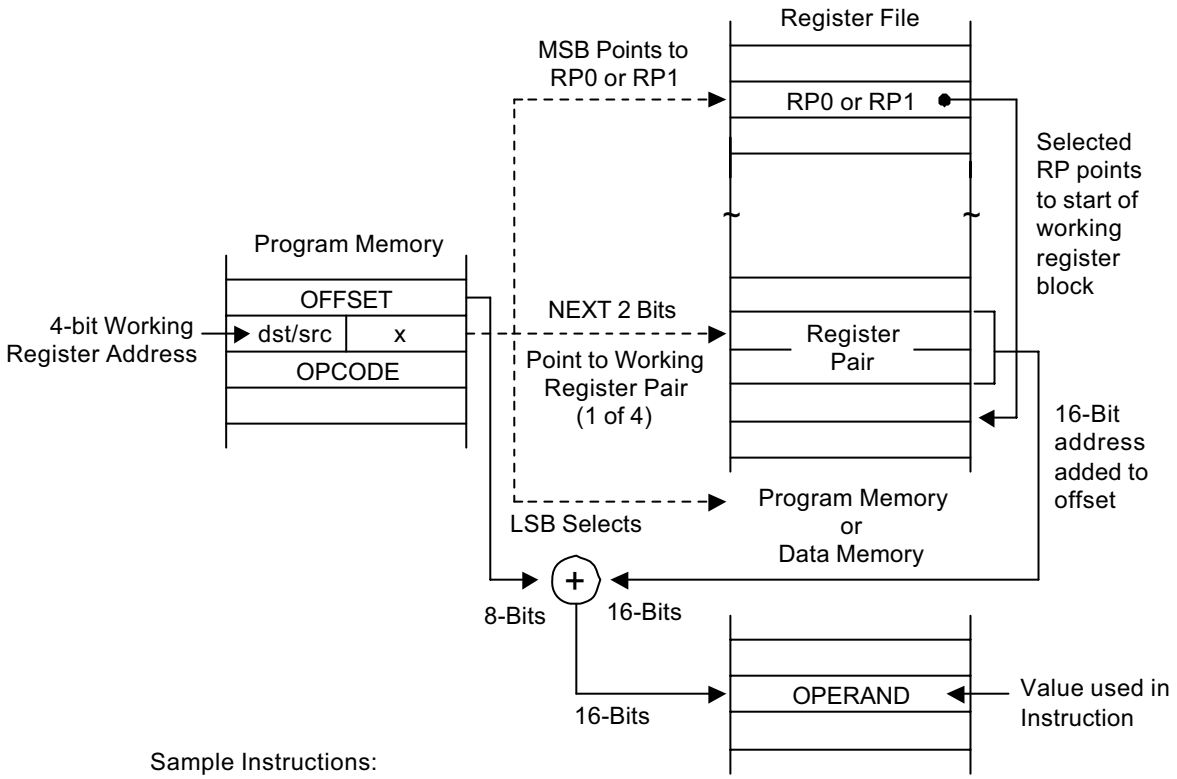


Figure 3-7. Indexed Addressing to Register File

INDEXED ADDRESSING MODE (Continued)

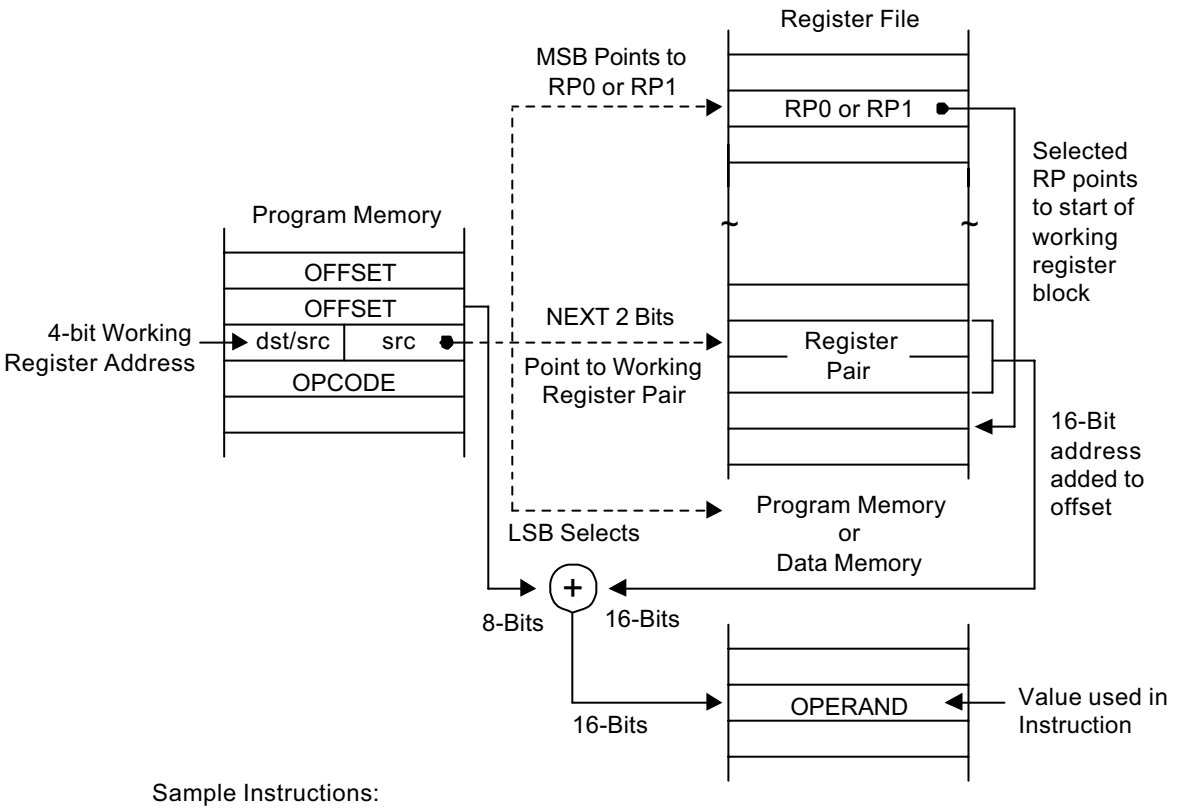


Sample Instructions:

```
LDC     R4, #04H[RR2]     ; The values in the program address (RR2 + 04H)
                          ; are loaded into register R4.
LDE     R4,#04H[RR2]      ; Identical operation to LDC example, except that
                          ; external program memory is accessed.
```

Figure 3-8. Indexed Addressing to Program or Data Memory with Short Offset

INDEXED ADDRESSING MODE (Concluded)



Sample Instructions:

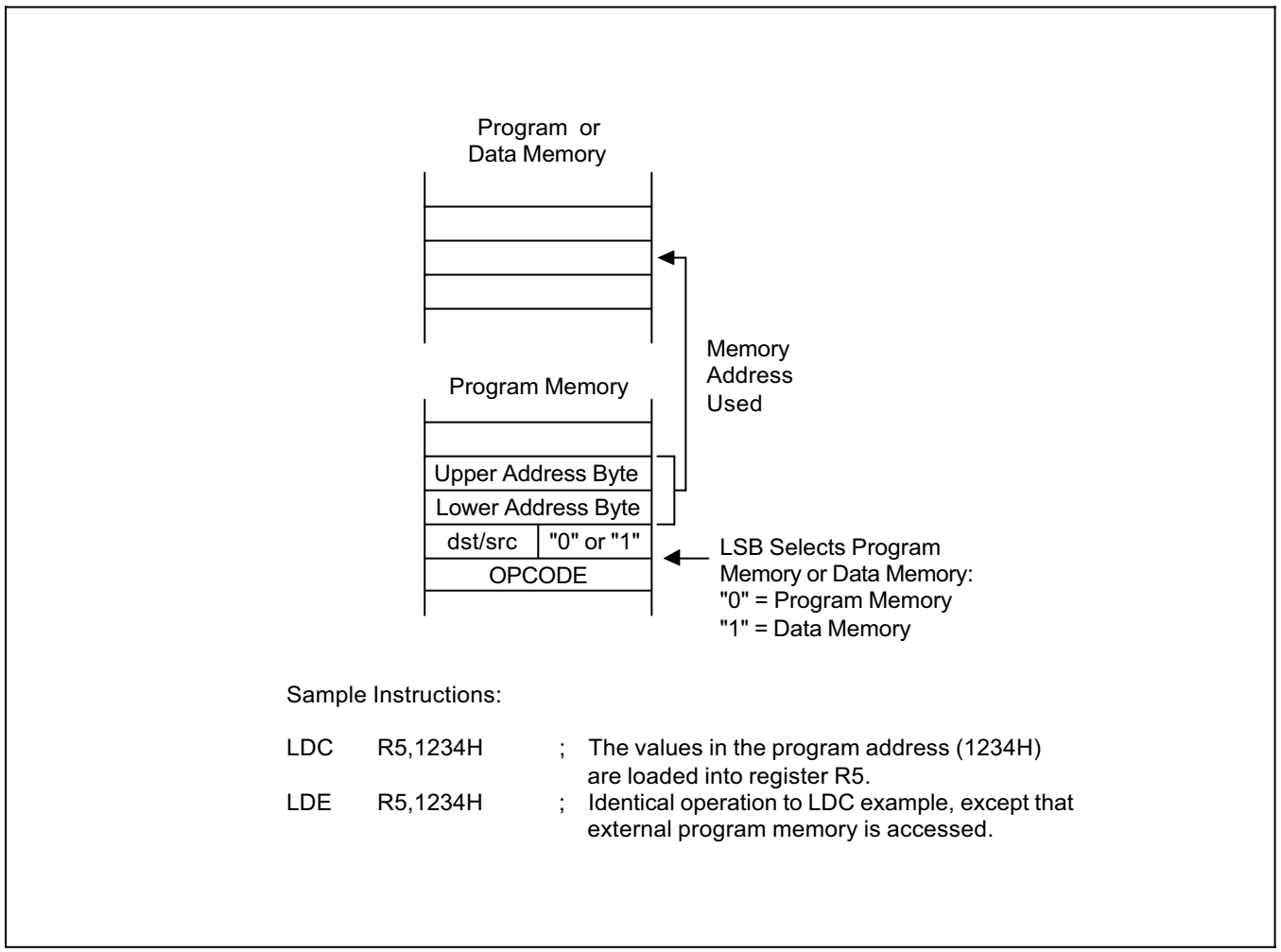
```
LDC     R4, #1000H[RR2]     ; The values in the program address (RR2 + 1000H)
                            ; are loaded into register R4.
LDE     R4,#1000H[RR2]      ; Identical operation to LDC example, except that
                            ; external program memory is accessed.
```

Figure 3-9. Indexed Addressing to Program or Data Memory

DIRECT ADDRESS MODE (DA)

In Direct Address (DA) mode, the instruction provides the operand's 16-bit memory address. Jump (JP) and Call (CALL) instructions use this addressing mode to specify the 16-bit destination address that is loaded into the PC whenever a JP or CALL instruction is executed.

The LDC and LDE instructions can use Direct Address mode to specify the source or destination address for Load operations to program memory (LDC) or to external data memory (LDE), if implemented.

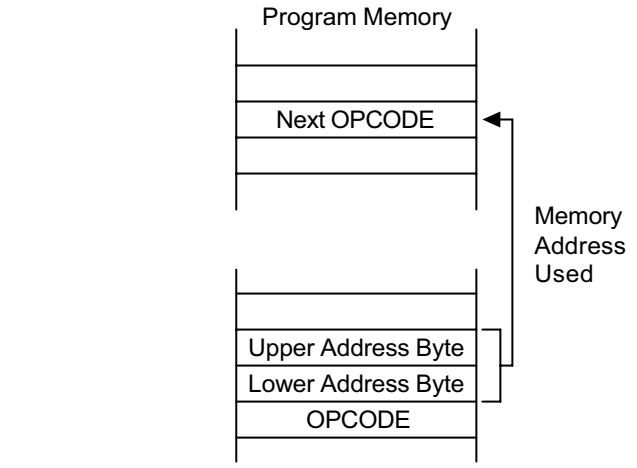


Sample Instructions:

```
LDC     R5,1234H    ;  The values in the program address (1234H)
                       are loaded into register R5.
LDE     R5,1234H    ;  Identical operation to LDC example, except that
                       external program memory is accessed.
```

Figure 3-10. Direct Addressing for Load Instructions

DIRECT ADDRESS MODE (Continued)



Sample Instructions:

```
JP      C,JOB1          ;  Where JOB1 is a 16-bit immediate address
CALL    DISPLAY         ;  Where DISPLAY is a 16-bit immediate address
```

Figure 3-11. Direct Addressing for Call and Jump Instructions

INDIRECT ADDRESS MODE (IA)

In Indirect Address (IA) mode, the instruction specifies an address located in the lowest 256 bytes of the program memory. The selected pair of memory locations contains the actual address of the next instruction to be executed. Only the CALL instruction can use the Indirect Address mode.

Because the Indirect Address mode assumes that the operand is located in the lowest 256 bytes of program memory, only an 8-bit address is supplied in the instruction; the upper bytes of the destination address are assumed to be all zeros.

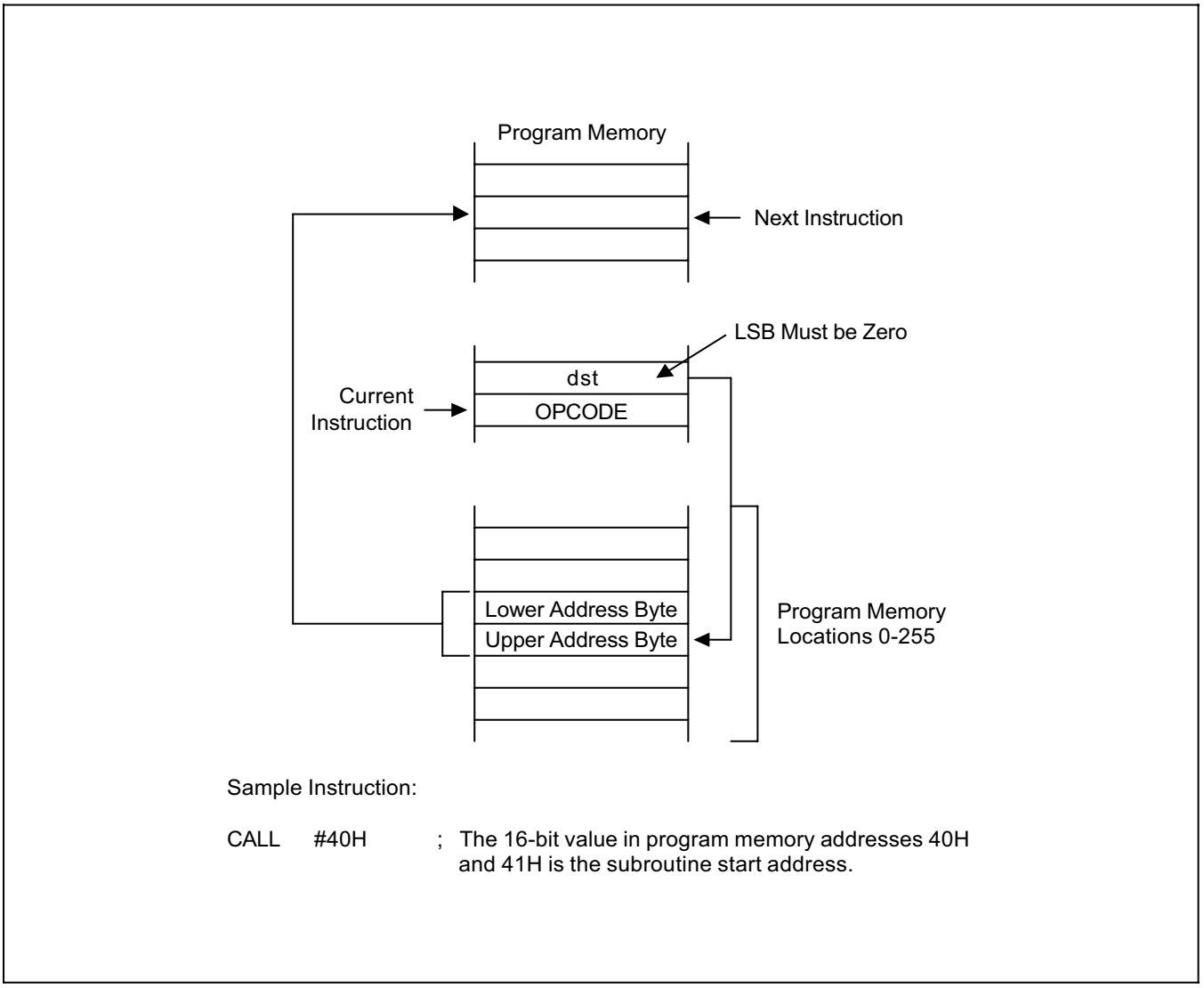


Figure 3-12. Indirect Addressing

RELATIVE ADDRESS MODE (RA)

In Relative Address (RA) mode, a twos-complement signed displacement between – 128 and + 127 is specified in the instruction. The displacement value is then added to the current PC value. The result is the address of the next instruction to be executed. Before this addition occurs, the PC contains the address of the instruction immediately following the current instruction.

The instructions that support RA addressing is JR.

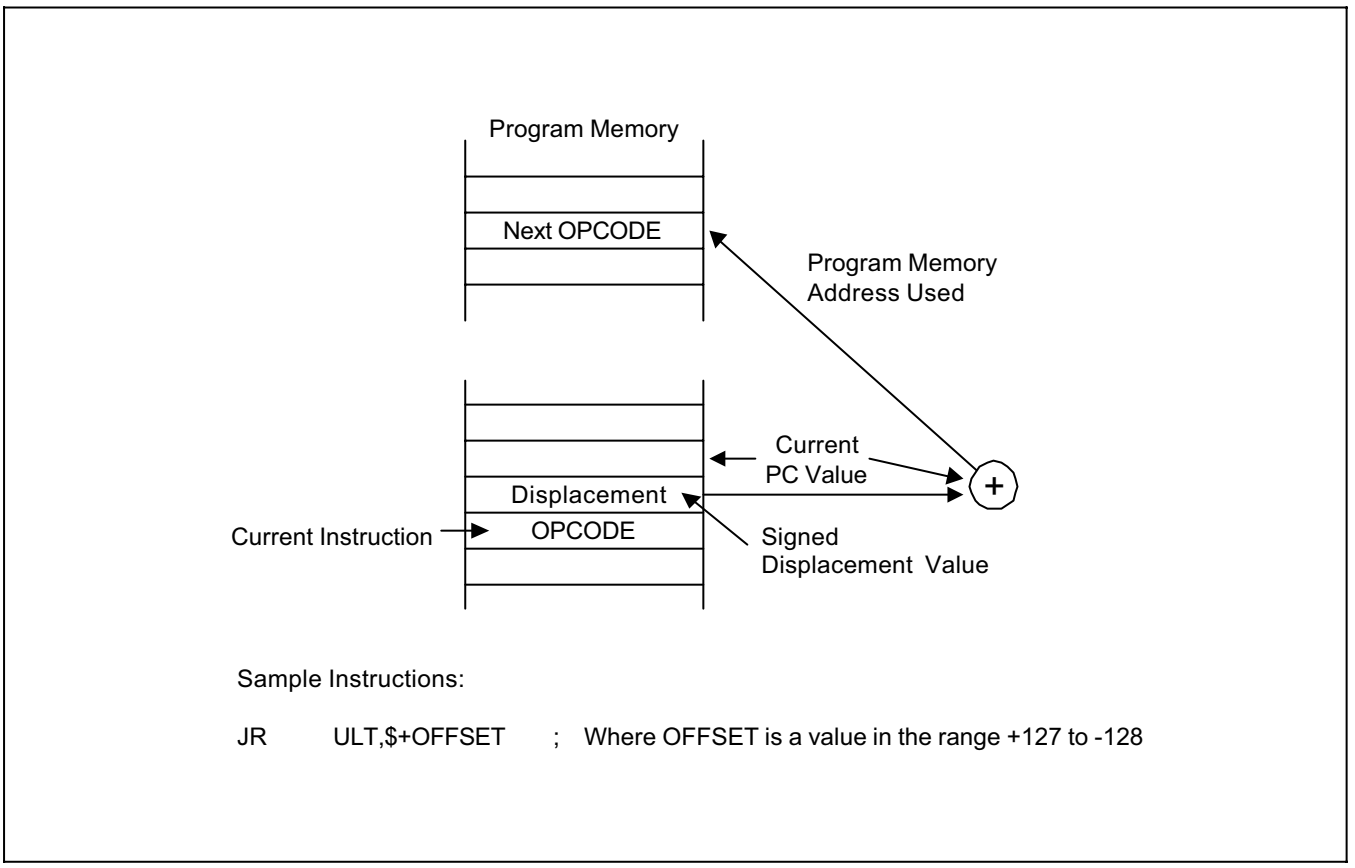


Figure 3-13. Relative Addressing

IMMEDIATE MODE (IM)

In Immediate (IM) addressing mode, the operand value used in the instruction is the value supplied in the operand field itself. Immediate addressing mode is useful for loading constant values into registers.

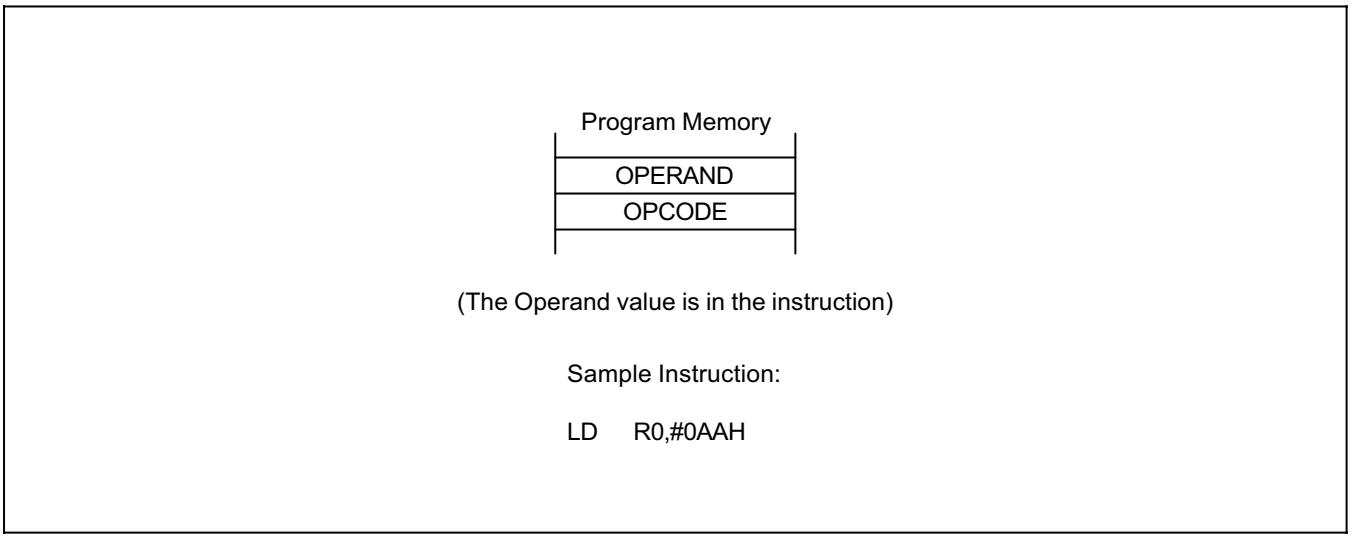


Figure 3-14. Immediate Addressing

4 CONTROL REGISTERS

OVERVIEW

Control register descriptions are arranged in alphabetical order according to register mnemonic. More detailed information about control registers is presented in the context of the specific peripheral hardware descriptions in Part II of this manual.

The locations and read/write characteristics of all mapped registers in the S3C9484/C9488/F9488 register file are listed in Table 4-1. The hardware reset value for each mapped register is described in Chapter 8, "RESET and Power-Down."

Table 4-1. System and Peripheral Registers

Register Name	Mnemonic	Decimal	Hex	R/W
LCD control register	LCDCON	208	D0H	R/W
LCD drive voltage control register	LCDVOL	209	D1H	R/W
Port 0 pull-up resistor control register	P0PUR	210	D2H	R/W
Port 1 pull-up resistor control register	P1PUR	211	D3H	R/W
System Clock control register	CLKCON	212	D4H	R/W
System flags register	FLAGS	213	D5H	R/W
Oscillator control register	OSCCON	214	D6H	R/W
STOP control register	STPCON	215	D7H	R/W
Voltage Level Detector control register	VLDCON	216	D8H	R/W
Stack pointer register	SP	217	D9H	R/W
Location DAH - DBH are not mapped				
Basic timer control register	BTCON	220	DCH	R/W
Basic timer counter register	BTCNT	221	DDH	R
Location DEH is not mapped				
System mode register	SYM	223	DFH	R/W

Table 4-1. System and Peripheral Registers (continued)

Register Name	Mnemonic	Decimal	Hex	R/W
Port 0 Data Register	P0	224	E0H	R/W
Port 1 Data Register	P1	225	E1H	R/W
Port 2 Data Register	P2	226	E2H	R/W
Port 3 Data Register	P3	227	E3H	R/W
Port 4 Data Register	P4	228	E4H	R/W
Watchdog timer control register	WDTCON	229	E5H	R/W
Port 0 control High register	P0CONH	230	E6H	R/W
Port 0 control Low register	P0CONL	231	E7H	R/W
Port 1 control High register	P1CONH	232	E8H	R/W
Port 1 control Low register	P1CONL	233	E9H	R/W
Port 2 control High register	P2CONH	234	EAH	R/W
Port 2 control Low register	P2CONL	235	EBH	R/W
Port 3 control High register	P3CONH	236	ECH	R/W
Port 3 control Low register	P3CONL	237	EDH	R/W
Port 3 interrupt control register	P3INT	238	EEH	R/W
Port 3 interrupt pending register	P3PND	239	EFH	R/W
Port 4 control High register	P4CONH	240	F0H	R/W
Port 4 control Low register	P4CONL	241	F1H	R/W
Timer A/Timer B interrupt pending register	TINTPND	242	F2H	RW
Timer A control register	TACON	243	F3H	R/W
Timer A counter register	TACNT	244	F4H	R
Timer A data register	TADATA	245	F5H	R/W
Timer B data register(high byte)	TBDATAH	246	F6H	R/W
Timer B data register(low byte)	TBDATAL	247	F7H	R/W
Timer B control register	TBCON	248	F8H	R/W
Watch timer control register	WTCON	249	F9H	R/W
A/D converter data register(high byte)	ADDATAH	250	FAH	R
A/D converter data register(low byte)	ADDATAL	251	FBH	R
A/D converter control register	ADCON	252	FCH	R/W
UART control register	UARTCON	253	FDH	R/W
UART pending register	UARTPND	254	FEH	R/W
UART data register	UDATA	255	FFH	R/W

Table 4-2. LCD display Register and Peripheral Registers (page 1)

Register Name	Mnemonic	Decimal	Hex	R/W
LCD Display RAM	–	0	00H	R/W
LCD Display RAM	–	1	01H	R/W
LCD Display RAM	–	2	02H	R/W
LCD Display RAM	–	3	03H	R/W
LCD Display RAM	–	4	04H	R/W
LCD Display RAM	–	5	05H	R/W
LCD Display RAM	–	6	06H	R/W
LCD Display RAM	–	7	07H	R/W
LCD Display RAM	–	8	08H	R/W
LCD Display RAM	–	9	09H	R/W
LCD Display RAM	–	10	0AH	R/W
LCD Display RAM	–	11	0BH	R/W
LCD Display RAM	–	12	0CH	R/W
LCD Display RAM	–	13	0DH	R/W
LCD Display RAM	–	14	0EH	R/W
LCD Display RAM	–	15	0FH	R/W
LCD Display RAM	–	16	10H	R/W
LCD Display RAM	–	17	11H	R/W
LCD Display RAM	–	18	12H	R/W
Location 13H is not mapped				
UART baud rate data register (high byte)	BRDATAH	20	14H	R/W
UART baud rate data register (low byte)	BRDATAL	21	15H	R/W

NOTE: When you use the SK-1000(SK-8xx) MDS , the BRDATAH/BRDATAL of mnemonic isn't showed on the system register window of MDS application program, because BRDATAH/BRDATAL is located on the general register page1.

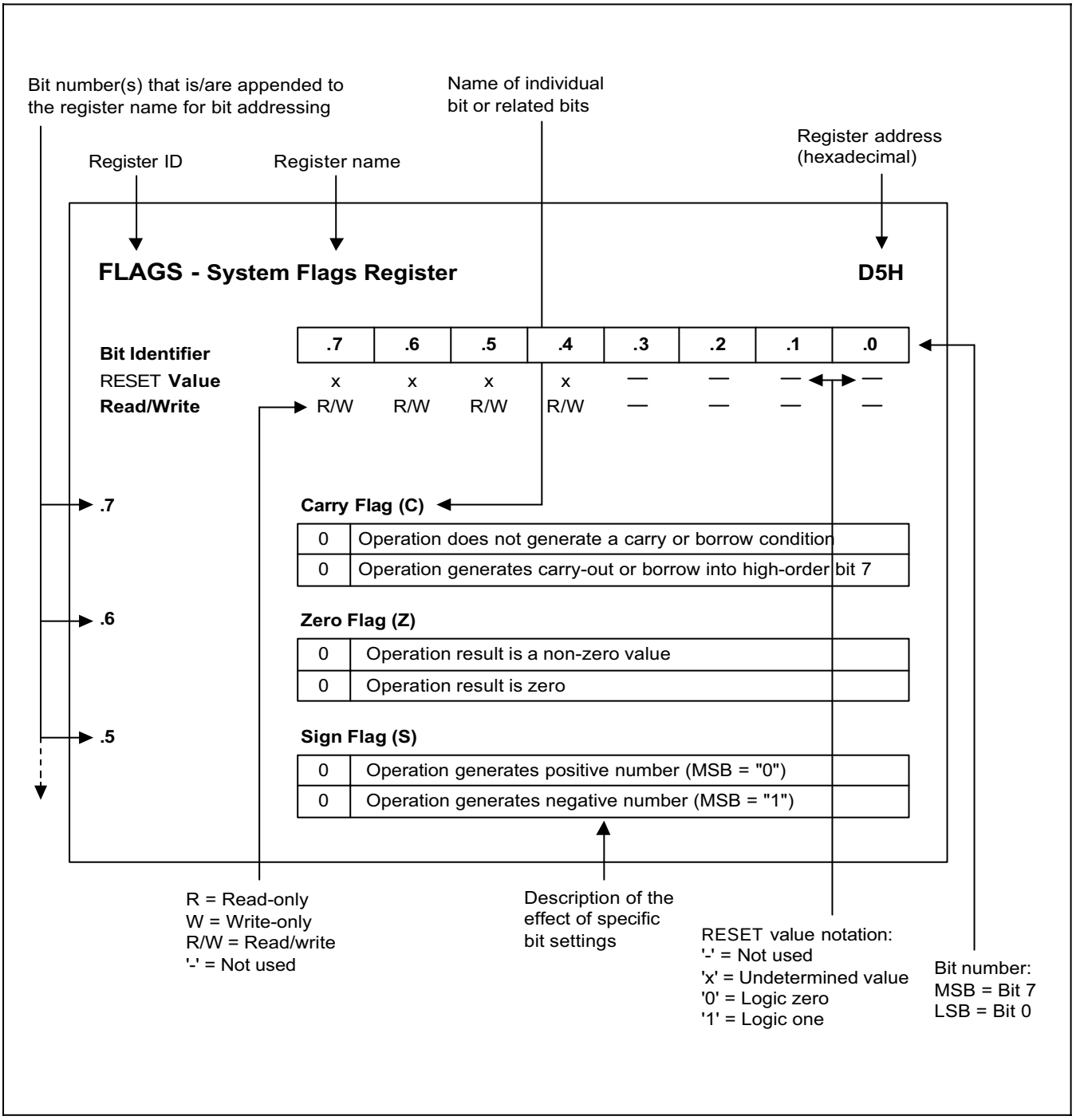

Bit number(s) that is/are appended to the register name for bit addressing
Name of individual bit or related bits
Register address (hexadecimal)
Register ID
Register name
FLAGS - System Flags Register
D5H
Bit Identifier
RESET Value
Read/Write
.7 .6 .5 .4 .3 .2 .1 .0
x x x x — — — —
R/W R/W R/W R/W — — — —
.7
Carry Flag (C)
0 Operation does not generate a carry or borrow condition
0 Operation generates carry-out or borrow into high-order bit 7
.6
Zero Flag (Z)
0 Operation result is a non-zero value
0 Operation result is zero
.5
Sign Flag (S)
0 Operation generates positive number (MSB = "0")
0 Operation generates negative number (MSB = "1")
R = Read-only
W = Write-only
R/W = Read/write
'-' = Not used
Description of the effect of specific bit settings
RESET value notation:
'-' = Not used
'x' = Undetermined value
'0' = Logic zero
'1' = Logic one
Bit number:
MSB = Bit 7
LSB = Bit 0

Figure 4-1. Register Description Format

ADCON — A/D Converter Control Register FCH

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7-.4 A/D Input Pin Selection Bits

0	0	0	0	ADC0
0	0	0	1	ADC1
0	0	1	0	ADC2
0	0	1	1	ADC3
0	1	0	0	ADC4
0	1	0	1	ADC5
0	1	1	0	ADC6
0	1	1	1	ADC7
1	0	0	0	ADC8
Other value				Connected with GND internally

.3 End-Of-Conversion (EOC) Status Bit

0	A/D conversion is in progress
1	A/D conversion complete

.2-.1 Clock Source Selection Bits

0	0	fxx/16 (fosc≤8MHz)
0	1	fxx/8 (fosc ≤ 8MHz)
1	0	fxx/4 (fosc≤ 8MHz)
1	1	fxx (fosc≤ 2.5MHz)

.0 A/D conversion Start Bit

0	Disable operation
1	Start operation

NOTE: Maximum ADC clock input = 4MHz.

BTCON — Basic Timer Control Register DCH

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	–	–	–	–	0	0	0	0
Read/Write	–	–	–	–	R/W	R/W	R/W	R/W

.7-.4

Not used for the S3C9484/C9488/F9488

.3-.2 Basic Timer Input Clock Selection Bits

0	0	fxx/4096 (3)
0	1	fxx/1024
1	0	fxx/128
1	1	Not used

.1 Basic Timer Counter Clear Bit (1)

0	No effect
1	Clear the basic timer counter value

.0 Clock Frequency Divider Clear Bit for Basic Timer (2)

0	No effect
1	Clear both clock frequency dividers

NOTES:

1. When you write a “1” to BTCON.1, the basic timer counter value is cleared to "00H". Immediately following the write operation, the BTCON.1 value is automatically cleared to “0”.
2. When you write a "1" to BTCON.0, the corresponding frequency divider is cleared to "00H". Immediately following the write operation, the BTCON.0 value is automatically cleared to "0".
3. The fxx is selected clock for system (main OSC. or sub OSC).

CLKCON — System Clock Control Register D4H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	–	–	0	0	–	–	–
Read/Write	R/W	–	–	R/W	R/W	–	–	–

.7 Oscillator IRQ Wake-up Function Enable Bit

0	Enable IRQ for main system oscillator wake-up function
1	Disable IRQ for main system oscillator wake-up function

.6-.5

Not used for the S3C9484/C9488/F9488

.4-.3 CPU Clock (System Clock) Selection Bits (note)

0	0	fxx/16
0	1	fxx/8
1	0	fxx/2
1	1	fxx/1 (non-divided)

.2-.0

Not used for the S3C9484/C9488/F9488

NOTE: After a reset, the slowest clock (divided by 16) is selected as the system clock. To select faster clock speeds, load the appropriate values to CLKCON.3 and CLKCON.4.

FLAGS — System Flags Register D5H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET /Value	x	x	x	x	–	–	–	–
Read/Write	R/W	R/W	R/W	R/W	–	–	–	–

.7 Carry Flag (C)

0	Operation does not generate a carry or borrow condition
1	Operation generates a carry-out or borrow into high-order bit 7

.6 Zero Flag (Z)

0	Operation result is a non-zero value
1	Operation result is zero

.5 Sign Flag (S)

0	Operation generates a positive number (MSB = "0")
1	Operation generates a negative number (MSB = "1")

.4 Overflow Flag (V)

0	Operation result is ≤ + 127 or ≥ – 128
1	Operation result is > + 127 or < – 128

.3–.0 Not used for the S3C9484/C9488/F9488

LCDCON — LCD Control Register D0H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	–	0	0	0	0	0	0
Read/Write	R/W	–	R/W	R/W	R/W	R/W	R/W	R/W

.7 LCD Module enable/disable Bit

0	Disable LCD Module
1	Enable LCD Module

.6

Not used for the S3C9484/C9488/F9488

.5-.4 LCD Duty Selection Bit

0	0	1/8 duty , 1/4 bias
0	1	1/4 duty , 1/3 bias
1	x	Static

.3-.2 LCD Dot On/Off Control Bits

0	0	Off signal
0	1	On signal
1	x	Normal display

.1-.0 LCD Clock Signal Selection Bits

0	0	$fw/2^7$
0	1	$fw/2^6$
1	0	$fw/2^5$
1	1	$fw/2^4$

LCDVOL — LCD Voltage Control Register D1H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	–	–	–	0	0	0	0
Read/Write	R/W	–	–	–	R/W	R/W	R/W	R/W

.7 **LCD Contrast Control Enable/Disable Bit**

0	Disable LCD Contrast Module
1	Enable LCD Contrast Module

.6-.4

Not used for the S3C9484/C9488/F9488

.3-.0 **LCD Segment/Port Output Selection Bits:**

0	0	0	0	1/16 step (The dimmest level)
0	0	0	1	2/16 step
0	0	1	0	3/16 step
0	0	1	1	4/16 step
0	1	0	0	5/16 step
0	1	0	1	6/16 step
0	1	1	0	7/16 step
0	1	1	1	8/16 step
1	0	0	0	9/16 step
1	0	0	1	10/16 step
1	0	1	0	11/16 step
1	0	1	1	12/16 step
1	1	0	0	13/16 step
1	1	0	1	14/16 step
1	1	1	0	15/16 step
1	1	1	1	16/16 step

OSCCON — Oscillator Control Register D6H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	–	–	–	–	0	0	–	0
Read/Write	–	–	–	–	R/W	R/W	–	R/W

.7-.4

Not used for the S3C9484/C9488/F9488

.3 Main System Oscillator Control Bit

0	Main System Oscillator RUN
1	Main System Oscillator STOP

.2 Sub System Oscillator Control Bit

0	Sub system oscillator RUN
1	Sub system oscillator STOP

.1

Not used for the S3C9484/C9488/F9488

.0 System Clock Selection Bit

0	Main oscillator select
1	Subsystem oscillator select

P0CONH — Port 0 Control Register (High Byte) E6H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7-.6 **P0.7/COM4/ADC4**

0	0	Input mode
0	1	Alternative function; ADC4 input
1	0	Push-pull output
1	1	Alternative function; LCD COM4 signal output

.5-.4 **P0.6/COM5/ADC5**

0	0	Input mode
0	1	Alternative function; ADC5 input
1	0	Push-pull output
1	1	Alternative function; LCD COM5 signal output

.3–.2 **P0.5/ COM6/ADC6**

0	0	Input mode
0	1	Alternative function; ADC6 input
1	0	Push-pull output
1	1	Alternative function; LCD COM6 signal output

.1–.0 **P0.4/ COM7/ADC7**

0	0	Input mode
0	1	Alternative function; ADC7 input
1	0	Push-pull output
1	1	Alternative function; LCD COM7 signal output

NOTE: When users use Port 0, users must be care of the pull-up resistance status.

P0CONL — Port 0 Control Register (Low Byte) E7H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7–.6 **P0.3/ADC8**

0	x	Input mode
1	0	Push-pull output
1	x	Alternative function; ADC8 input

.5–.4 **P0.2**

0	x	Input mode
1	x	Push-pull output

.3–.2 **P0.1**

0	x	Input mode
1	x	Push-pull output

.1–.0 **P0.0**

0	x	Input mode
1	x	Push-pull output

NOTES:
1. If you selected the Xtin/Xtout function at Smart option, no relations to P0CONL.3 -.0 bit value. But if you selected the normal I/O function at Smart option, the reset value of P0CONL.3 -.0 bits are '0000'.
2. When users use Port 0, users must be care of the pull-up resistance status.

P0PUR — Port 0 Pull-up Resistor Control Register D2H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7 P0.7 Pull-up Resistor Enable/Disable

0	Pull-up resistor disable
1	Pull-up resistor enable

.6 P0.6 Pull-up Resistor Enable/Disable

0	Pull-up resistor disable
1	Pull-up resistor enable

.5 P0.5 Pull-up Resistor Enable/Disable

0	Pull-up resistor disable
1	Pull-up resistor enable

.4 P0.4 Pull-up Resistor Enable/Disable

0	Pull-up resistor disable
1	Pull-up resistor enable

.3 P0.3 Pull-up Resistor Enable/Disable

0	Pull-up resistor disable
1	Pull-up resistor enable

.2 P0.2 Pull-up Resistor Enable/Disable

0	Pull-up resistor disable
1	Pull-up resistor enable

.1 P0.1 Pull-up Resistor Enable/Disable

0	Pull-up resistor disable
1	Pull-up resistor enable

.0 P0.0 Pull-up Resistor Enable/Disable

0	Pull-up resistor disable
1	Pull-up resistor enable

P1CONH — Port 1 Control Register (High Byte) E8H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7-.6 **P1.7/COM0**

0	x	Input mode
1	0	Push-pull output
1	1	Alternative function; LCD COM0 signal output

.5-.4 **P1.6/COM1**

0	x	Input mode
1	0	Push-pull output
1	1	Alternative function; LCD COM1 signal output

.3-.2 **P1.5/COM2**

0	x	Input mode
1	0	Push-pull output
1	1	Alternative function; LCD COM2 signal output

.1-.0 **P1.4/COM3**

0	x	Input mode
1	0	Push-pull output
1	1	Alternative function; LCD COM3 signal output

NOTE: When users use Port 1, users must be care of the pull-up resistance status.

P1CONL — Port 1 Control Register (Low Byte) E9H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7-.6 **P1.3/ADC0**

0	X	Input mode
1	0	Push-pull output
1	1	Alternative function; ADC0 input

.5-.4 **P1.2/ADC1**

0	X	Input mode
1	0	Push-pull output
1	1	Alternative function; ADC1 input

.3-.2 **P1.1/ADC2/BUZ**

0	0	Input mode
0	1	Alternative function; BUZ output
1	0	Push-pull output
1	1	Alternative function; ADC2 input

.1-.0 **P1.0/ADC3/TBPWM**

0	0	Input mode
0	1	Alternative function; TBPWM output
1	0	Push-pull output
1	1	Alternative function; ADC3 input

NOTE: When users use Port 1, users must be care of the pull-up resistance status.

P1PUR — Port 1 Pull-up Resistor Control Register D3H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7 P1.7 Pull-up Resistor Enable/Disable

0	Pull-up resistor disable
1	Pull-up resistor enable

.6 P1.6 Pull-up Resistor Enable/Disable

0	Pull-up resistor disable
1	Pull-up resistor enable

.5 P1.5 Pull-up Resistor Enable/Disable

0	Pull-up resistor disable
1	Pull-up resistor enable

.4 P1.4 Pull-up Resistor Enable/Disable

0	Pull-up resistor disable
1	Pull-up resistor enable

.3 P1.3 Pull-up Resistor Enable/Disable

0	Pull-up resistor disable
1	Pull-up resistor enable

.2 P1.2 Pull-up Resistor Enable/Disable

0	Pull-up resistor disable
1	Pull-up resistor enable

.1 P1.1 Pull-up Resistor Enable/Disable

0	Pull-up resistor disable
1	Pull-up resistor enable

.0 P1.0 Pull-up Resistor Enable/Disable

0	Pull-up resistor disable
1	Pull-up resistor enable

P2CONH — Port 2 Control Register (High Byte) EAH

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7–.6 **P2.7/SEG10**

0	0	Input mode with pull-up
0	1	Input mode
1	0	Push-pull output
1	1	Alternative function; LCD SEG10 signal output

.5-.4 **P2.6/SEG9**

0	0	Input mode with pull-up
0	1	Input mode
1	0	Push-pull output
1	1	Alternative function; LCD SEG9 signal output

.3–.2 **P2.5/SEG8**

0	0	Input mode with pull-up
0	1	Input mode
1	0	Push-pull output
1	1	Alternative function; LCD SEG8 signal output

.1–.0 **P2.4/SEG7**

0	0	Input mode with pull-up
0	1	Input mode
1	0	Push-pull output
1	1	Alternative function; LCD SEG7 signal output

P2CONL — Port 2 Control Register (Low Byte) EBH

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7-.6 **P2.3/SEG6**

0	0	Input mode with pull-up
0	1	Input mode
1	0	Push-pull output
1	1	Alternative function; LCD SEG6 signal output

.5-.4 **P2.2/SEG5**

0	0	Input mode with pull-up
0	1	Input mode
1	0	Push-pull output
1	1	Alternative function; LCD SEG5 signal output

.3-.2 **P2.1/SEG4**

0	0	Input mode with pull-up
0	1	Input mode
1	0	Push-pull output
1	1	Alternative function; LCD SEG4 signal output

.1-.0 **P2.0/SEG3**

0	0	Input mode with pull-up
0	1	Input mode
1	0	Push-pull output
1	1	Alternative function; LCD SEG3 signal output

P3CONH — Port 3 Control Register (High Byte) ECH

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	S	S	S	S	S	S	S	S
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7–.6 **P3.6/TACAP/INT3**

0	0	Input mode with pull-up; interrupt(INT3) input; TACAP
0	1	Input mode; interrupt(INT3) input; TACAP
1	x	Push-pull output

.5–.4 **P3.5/TACK/INT2**

0	0	Input mode with pull-up; interrupt(INT2) input; TACK
0	1	Input mode; interrupt(INT2) input; TACK
1	x	Push-pull output

.3–.2 **P3.4/TAOUT(TAPWM)/INT1**

0	0	Input mode with pull-up; interrupt(INT1) input
0	1	Input mode; interrupt(INT1) input
1	0	Push-pull output
1	1	Alternative function; TAOUT(TAPWM)

.1–.0 **P3.3/SEG18/INT0**

0	0	Input mode with pull-up; interrupt(INT0) input
0	1	Input mode; interrupt(INT0) input
1	0	Push-pull output
1	1	Alternative function; LCD SEG18 signal output

NOTE: 'S' of reset value mean that reset value is set by smart option.

P3CONL — Port 3 Control Register (Low Byte) EDH

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7-.5 **P3.2/SEG17/TXD**

0	0	0	Input mode with pull-up
0	0	1	Input mode
0	1	0	Push-pull output
0	1	1	Alternative function; TXD output
1	x	x	Alternative function; LCD SEG17 signal output

.4-.2 **P3.1/SEG16/RXD**

0	0	0	Input mode with pull-up; RXD input
0	0	1	Input mode; RXD input
0	1	0	Push-pull output
0	1	1	Alternative function; RXD output
1	x	x	Alternative function; LCD SEG16 signal output

.1-.0 **P3.0/ SEG15**

0	0	Input mode with pull-up
0	1	Input mode
1	0	Push-pull output
1	1	Alternative function; LCD SEG15 signal output

P3INT — Port 3 Interrupt Control Register EEH

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7-.6 P3.6/ INT3 Interrupt Enable/Disable Selection Bits

0	x	Interrupt Disable
1	0	Interrupt Enable; falling edge
1	1	Interrupt Enable; rising edge

.5-.4 P3.5/ INT2 Interrupt Enable/Disable Selection Bits

0	x	Interrupt Disable
1	0	Interrupt Enable; falling edge
1	1	Interrupt Enable; rising edge

.3-.2 P3.4/ INT1 Interrupt Enable/Disable Selection Bits

0	x	Interrupt Disable
1	0	Interrupt Enable; falling edge
1	1	Interrupt Enable; rising edge

.1-.0 P3.3/INT0 Interrupt Enable/Disable Selection Bits

0	x	Interrupt Disable
1	0	Interrupt Enable; falling edge
1	1	Interrupt Enable; rising edge

P3PND — Port 3 Interrupt Pending Register EFH

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	–	–	–	–	0	0	0	0
Read/Write	–	–	–	–	R/W	R/W	R/W	R/W

.7-.4

Not used for the S3C9484/C9488/F9488

.3 P3.6/INT3 Interrupt Pending Bit

0	Interrupt request is not pending, pending bit clear when write 0
1	Interrupt request is pending

.2 P3.5/INT2 Interrupt Pending Bit

0	Interrupt request is not pending, pending bit clear when write 0
1	Interrupt request is pending

.1 P3.4/INT1 Interrupt Pending Bit

0	Interrupt request is not pending, pending bit clear when write 0
1	Interrupt request is pending

.0 P3.3/INT0 Interrupt Pending Bit

0	Interrupt request is not pending, pending bit clear when write 0
1	Interrupt request is pending

P4CONH — Port 4 Control Register (High Byte) F0H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	–	–	0	0	0	0	0	0
Read/Write	–	–	R/W	R/W	R/W	R/W	R/W	R/W

.7–.6 Not used for the S3C9484/C9488/F9488

.5-.4 **P4.6/SEG14**

0	0	Input mode with pull-up
0	1	Input mode
1	0	Push-pull output
1	1	Alternative function; LCD SEG14 signal output

.3–.2 **P4.5/SEG13**

0	0	Input mode with pull-up
0	1	Input mode
1	0	Push-pull output
1	1	Alternative function; LCD SEG13 signal output

.1–.0 **P4.4/SEG12**

0	0	Input mode with pull-up
0	1	Input mode
1	0	Push-pull output
1	1	Alternative function; LCD SEG12 signal output

P4CONL — Port 4 Control Register (Low Byte) F1H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7-.6 **P4.3/SEG11**

0	0	Input mode with pull-up
0	1	Input mode
1	0	Push-pull output
1	1	Alternative function; LCD SEG11 signal output

.5-.4 **P4.2/SEG2**

0	0	Input mode with pull-up
0	1	Input mode
1	0	Push-pull output
1	1	Alternative function; LCD SEG2 signal output

.3-.2 **P4.1/SEG1**

0	0	Input mode with pull-up
0	1	Input mode
1	0	Push-pull output
1	1	Alternative function; LCD SEG1signal output

.1-.0 **P4.0/SEG0**

0	0	Input mode with pull-up
0	1	Input mode
1	0	Push-pull output
1	1	Alternative function; LCD SEG0 signal output

SP — Stack Pointer D9H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	x	x	x	x	x	x	x	x
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7–.0 Stack Pointer Address

The stack pointer value is 8-bit stack pointer address (SP7–SP0). The SP value is undefined following a reset.

STPCON — Stop Control Register D7H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7–.0 STOP Control Bits

1 0 1 0 0 1 0 1	Enable stop instruction
Other values	Disable stop instruction

NOTE: Before executing the STOP instruction, you must set this STPCON register as “10100101b”. Otherwise the STOP instruction will not be executed.

SYM — System Mode Register DFH

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	–	–	–	–	0	0	0	0
Read/Write	–	–	–	–	R/W	R/W	R/W	R/W

.7–.4

Not used for S3C9484/C9488/F9488

.3 Global Interrupt Enable Bit

0	Disable all interrupts
1	Enable all interrupt

.2–.0 Page Select Bits

0	0	0	Page 0
0	0	1	Page 1
0	1	0	Page 2 (Not used for S3C9484/C9488/F9488)
0	1	1	Page 3 (Not used for S3C9484/C9488/F9488)
1	0	0	Page 4 (Not used for S3C9484/C9488/F9488)
1	0	1	Page 5 (Not used for S3C9484/C9488/F9488)
1	1	0	Page 6 (Not used for S3C9484/C9488/F9488)
1	1	1	Page 7 (Not used for S3C9484/C9488/F9488)

NOTE: Following a reset, you must enable global interrupt processing by executing an EI instruction (not by writing a "1" to SYM.3).

TACON — Timer A Control Register F3H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7-.6 Timer A Input Clock Selection Bits

0	0	fxx/1024
0	1	fxx/256
1	0	fxx/64
1	1	External clock (TACK)

.5-.4 Timer A Operating Mode Selection Bits

0	0	Internal mode (TAOUT mode)
0	1	Capture mode (capture on rising edge, counter running, OVF can occur)
1	0	Capture mode (capture on falling edge, counter running, OVF can occur)
1	1	PWM mode (OVF interrupt can occur)

.3 Timer A Counter Clear Bit

0	No effect
1	Clear the timer A counter (After clearing, return to zero)

.2 Timer A Overflow Interrupt Enable Bit

0	Disable interrupt
1	Enable interrupt

.1 Timer A Match/Capture Interrupt Enable Bit

0	Disable interrupt
1	Enable interrupt

.0 Timer A Start/Stop Bit

0	Stop Timer A
1	Start Timer A

TBCON — Timer B Control Register F8H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7–.6 Timer B Input Clock Selection Bits

0	0	fxx
0	1	fxx/2
1	0	fxx/4
1	1	fxx/8

.5–.4 Timer B Interrupt Time Selection Bits

0	0	Elapsed time for low data value
0	1	Elapsed time for high data value
1	0	Elapsed time for low and high data values
1	1	Invalid setting

.3 Timer B Underflow Interrupt Enable Bit

0	Disable Interrupt
1	Enable Interrupt

.2 Timer B Start/Stop Bit

0	Stop timer B
1	Start timer B

.1 Timer B Mode Selection Bit

0	One-shot mode
1	Repeating mode

.0 Timer B Output flip-flop Control Bit

0	T-FF is low
1	T-FF is high

NOTE: fxx is selected clock for system.

TINTPND — Timer A,B Interrupt Pending Register F2H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	–	–	–	–	–	0	0	0
Read/Write	–	–	–	–	–	R/W	R/W	R/W

.7-.3 Not used for the S3C9484/C9488/F9488

.2 Timer B Underflow Interrupt Pending Bit

0	No interrupt pending
0	*Clear pending bit when write*
1	Interrupt pending

.1 Timer A Overflow Interrupt Pending Bit

0	No interrupt pending
0	*Clear pending bit when write*
1	Interrupt pending

.0 Timer A Match/Capture Interrupt Pending Bit

0	No interrupt pending
0	*Clear pending bit when write*
1	Interrupt pending

UARTCON — UART Control Register FDH

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7–.6 Operating mode and baud rate selection bits

0	0	Mode 0: Shift Register [fxx/(16 × (16bit BRDATA + 1))]
0	1	Mode 1: 8-bit UART [fxx/(16 × (16bit BRDATA + 1))]
1	x	Mode 2: 9-bit UART [fxx/(16 × (16bit BRDATA + 1))]

.5 Multiprocessor communication(1) enable bit (for modes 2 only)

0	Disable
1	Enable

.4 Serial data receive enable bit

0	Disable
1	Enable

.3

If Parity disable mode (PEN = 0),
Location of the 9th data bit to be transmitted in UART mode 2 ("0" or "1").

If Parity enable mode (PEN = 1),
even/odd parity selection bit for transmit data in UART mode 2.
0: Even parity bit generation for transmit data
1: Odd parity bit generation for transmit data

UARTCON — UART Control Register (Continued) FDH

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.2

If Parity disable (PEN = 0),
location of the 9th data bit that was received in UART mode 2 ("0" or "1").

If Parity enable mode (PEN = 1),
even/odd parity selection bit for receive data in UART mode 2.
0: Even parity check for the received data
1: Odd parity check for the received data

A result of parity error will be saved in RPE bit of the UARTPND register after parity checking of the received data.

.1 **Receive interrupt enable bit**

0	Disable Receive interrupt
1	Enable Receive interrupt

.0 **Transmit interrupt enable bit**

0	Disable Transmit interrupt
1	Enable Transmit Interrupt

NOTES:

1. In mode 2, if the MCE (UARTCON.5) bit is set to "1", the receive interrupt will not be activated if the received 9th data bit is "0". In mode 1, if MCE = "1", the receive interrupt will not be activated if a valid stop bit was not received. In mode 0, the MCE (UARTCON.5) bit should be "0".
2. The descriptions for 8-bit and 9-bit UART mode don't include start and stop bits for serial data receive and transmit.
3. Parity enable bits, PEN, are located in the UARTPND register at address FEH.
4. Parity enable and parity error check can be available in 9-bit UART mode (Mode 2) only.

UARTPND — UART Pending and parity control FEH

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	–	–	0	0	–	–	0	0
Read/Write	–	–	R/W	R/W	–	–	R/W	R/W

.7-.6 Not used for the S3C9484/C9488/F9488

.5 UART parity enable/disable (PEN)

0	Disable
1	Enable

.4 UART receive parity error (RPE)

0	No error
1	Parity error

.3-.2 Not used for the S3C9484/C9488/F9488

.1 UART receive interrupt pending flag

0	Not pending
0	*Clear pending bit (when write)*
1	Interrupt pending

.0 UART transmit interrupt pending flag

0	Not pending
0	*Clear pending bit (when write)*
1	Interrupt pending

NOTES:

1. In order to clear a data transmit or receive interrupt pending flag, you must write a "0" to the appropriate pending bit.
2. To avoid programming errors, we recommend using load instruction (except for LDB), when manipulating UARTPND values.
3. Parity enable and parity error check can be available in 9-bit UART mode (Mode 2) only.
4. Parity error bit (RPE) will be refreshed whenever 8th receive data bit has been shifted.

VLDCON — Voltage Level Detector Control Register D8H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	–	0	1	0	1	1	0	0
Read/Write	–	R	R/W	R/W	R/W	R/W	R/W	R/W

.7 Not used for the S3C9484/C9488/F9488

.6 VLD Level Set Bit

0	V_{DD} is higher than reference voltage
1	V_{DD} is lower than reference voltage

.5-.1 Reference Voltage Selection Bits

10110	V_{VLD} = 2.4 V
10011	V_{VLD} = 2.7 V
01110	V_{VLD} = 3.3 V
01011	V_{VLD} = 3.9 V
Other values	Don't care

.0 VLD Operation Enable Bit

0	Operation off
1	Operation on

WDTCON — Watchdog Timer Control Register E5H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7-.4 **Watchdog Timer Function Enable Bits (for System Reset)**

1	0	1	0	Disable watchdog timer function
Other values				Enable watchdog timer function

.3-.0 **Watchdog Timer Counter Clear Bits**

1	0	1	0	Clear watchdog timer counter
Other values				Don’t care

WTCON — Watch Timer Control Register F9H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7 Watch Timer Clock Selection Bit

0	Main system clock divided by 2^7 (fxx/128)
1	Sub system clock (fxt)

.6 Watch Timer Interrupt Enable Bit

0	Disable watch timer interrupt
1	Enable watch timer interrupt

.5–.4 Buzzer Signal Selection Bits

0	0	0.5 kHz buzzer (BUZ) signal output
0	1	1 kHz buzzer (BUZ) signal output
1	0	2 kHz buzzer (BUZ) signal output
1	1	4 kHz buzzer (BUZ) signal output

.3–.2 Watch Timer Speed Selection Bits

0	0	1.0 s Interval
0	1	0.5 s Interval
1	0	0.25 s Interval
1	1	3.91 ms Interval

.1 Watch Timer Enable Bit

0	Disable watch timer; Clear frequency dividing circuits
1	Enable watch timer

.0 Watch Timer Interrupt Pending Bit

0	Interrupt is not pending, Clear pending bit when write
1	Interrupt is pending

5 INTERRUPT STRUCTURE

OVERVIEW

The SAM88RCRI interrupt structure has two basic components: a vector, and sources. The number of interrupt sources can be serviced through an interrupt vector which is assigned in ROM address 0000H.

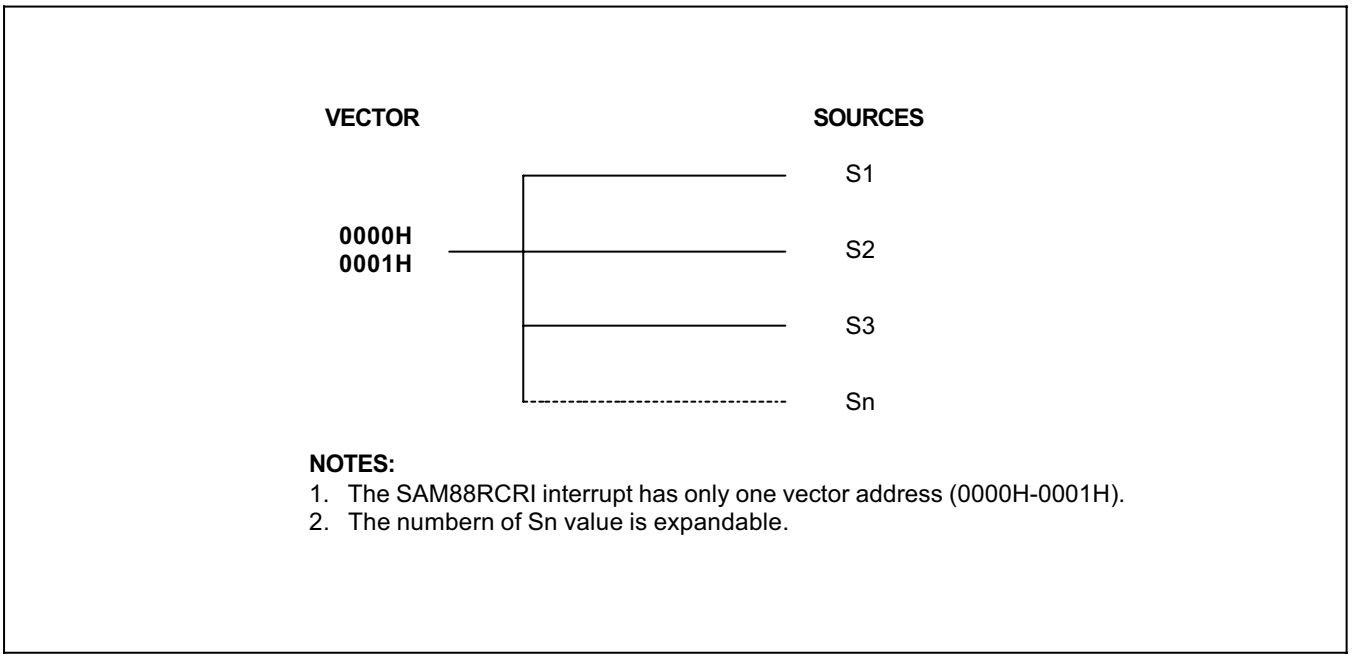


Figure 5-1. S3C9-Series Interrupt Type

INTERRUPT PROCESSING CONTROL POINTS

Interrupt processing can be controlled in two ways: either globally or specific interrupt level and source. The system-level control points in the interrupt structure are therefore:

— Global interrupt enable and disable (by EI and DI instructions)

— Interrupt source enable and disable settings in the corresponding peripheral control register(s)

ENABLE/DISABLE INTERRUPT INSTRUCTIONS (EI, DI)

The system mode register, SYM (DFH), is used to enable and disable interrupt processing.

SYM.3 is the enable and disable bit for global interrupt processing respectively, by modifying SYM.3. An Enable Interrupt (EI) instruction must be included in the initialization routine that follows a reset operation in order to enable interrupt processing. Although you can manipulate SYM.3 directly to enable and disable interrupts during normal operation, we recommend that you use the EI and DI instructions for this purpose.

INTERRUPT PENDING FUNCTION TYPES

When the interrupt service routine has executed, the application program's service routine must clear the appropriate pending bit before the return from interrupt subroutine (IRET) occurs.

INTERRUPT PRIORITY

Because there is not a interrupt priority register in SAM88RCRI, the order of service is determined by a sequence of source which is executed in interrupt service routine.

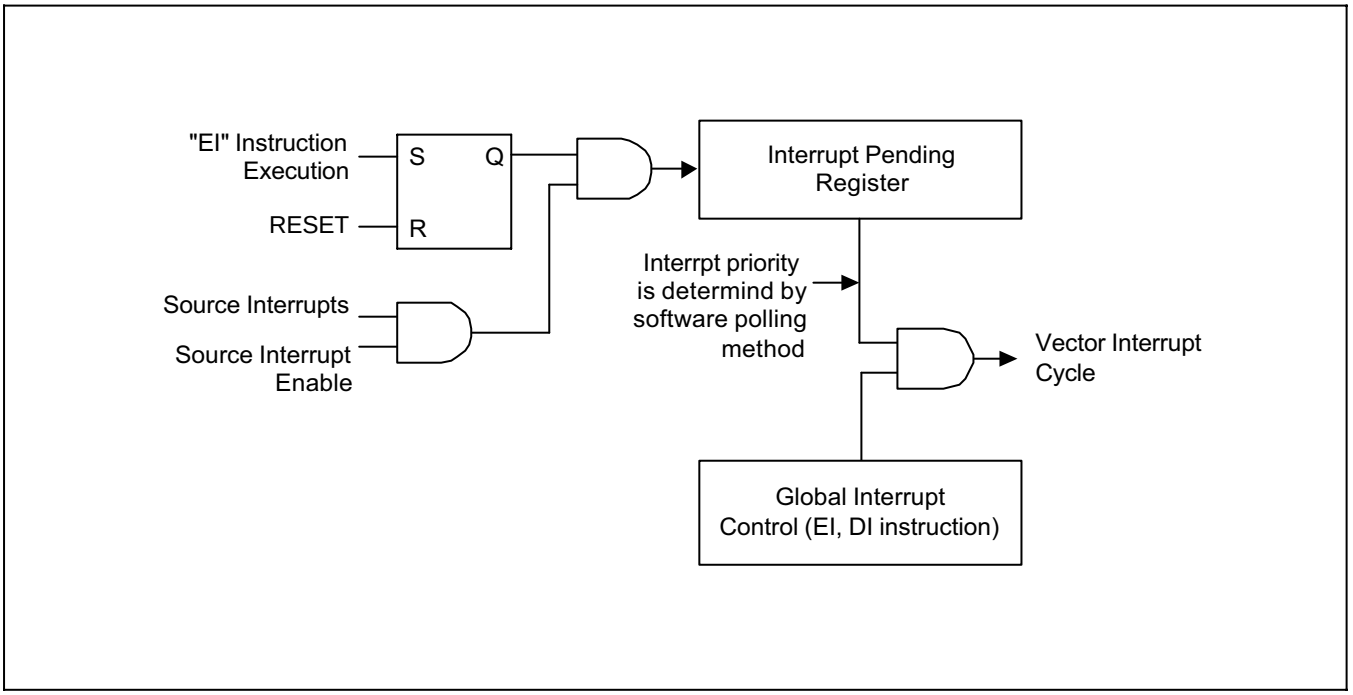


Figure 5-2. Interrupt Function Diagram

INTERRUPT SOURCE SERVICE SEQUENCE

The interrupt request polling and servicing sequence is as follows:

1. A source generates an interrupt request by setting the interrupt request pending bit to "1".
2. The CPU generates an interrupt acknowledge signal.
3. The service routine starts and the source's pending flag is cleared to "0" by software.
4. Interrupt priority must be determined by software polling method.

INTERRUPT SERVICE ROUTINES

Before an interrupt request can be serviced, the following conditions must be met:

— Interrupt processing must be enabled (EI, SYM.3 = "1")

— Interrupt must be enabled at the interrupt's source (peripheral control register)

If all of the above conditions are met, the interrupt request is acknowledged at the end of the instruction cycle. The CPU then initiates an interrupt machine cycle that completes the following processing sequence:

1. Reset (clear to "0") the global interrupt enable bit in the SYM register (DI, SYM.3 = "0") to disable all subsequent interrupts.
2. Save the program counter and status flags to stack.
3. Branch to the interrupt vector to fetch the service routine's address.
4. Pass control to the interrupt service routine.

When the interrupt service routine is completed, an Interrupt Return instruction (IRET) occurs. The IRET restores the PC and status flags and sets SYM.3 to "1" (EI), allowing the CPU to process the next interrupt request.

GENERATING INTERRUPT VECTOR ADDRESSES

The interrupt vector area in the ROM contains the address of the interrupt service routine. Vectored interrupt processing follows this sequence:

1. Push the program counter's low-byte value to stack.
2. Push the program counter's high-byte value to stack.
3. Push the FLAGS register values to stack.
4. Fetch the service routine's high-byte address from the vector address 0000H.
5. Fetch the service routine's low-byte address from the vector address 0001H.
6. Branch to the service routine specified by the 16-bit vector address.

S3C9484/C9488/F9488 INTERRUPT STRUCTURE

The S3C9484/C9488/F9488 microcontroller has four peripheral interrupt sources:

— Timer A match / overflow
— Timer B underflow
— P3.3 / P3.4 / P3.5 / P3.6 external interrupt
— Watch Timer interrupt
— UART transmit interrupt / receive interrupt

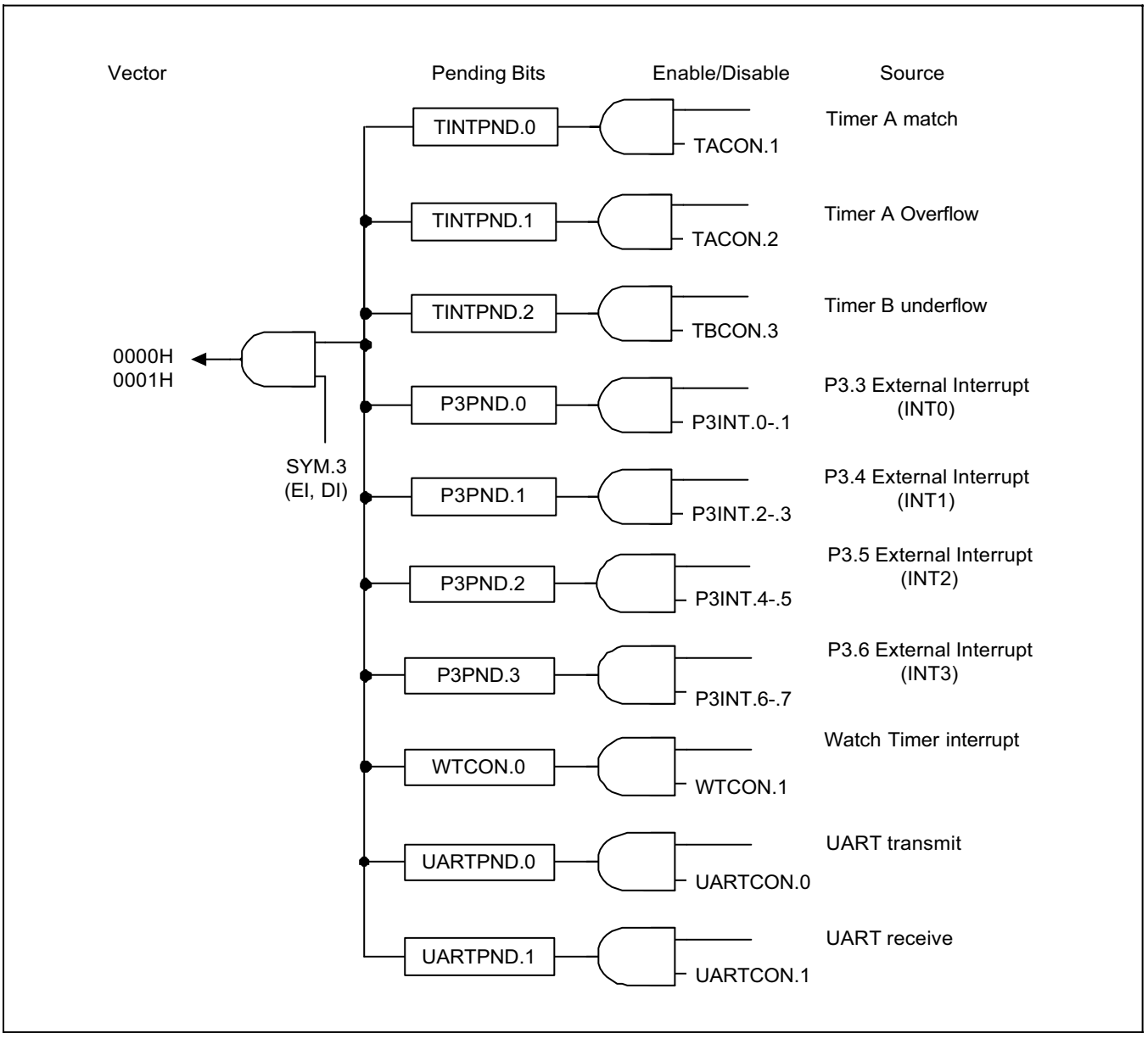


Figure 5-3. S3C9484/C9488/F9488 Interrupt Structure

6 SAM88RCRI INSTRUCTION SET

OVERVIEW

The SAM88RCRI instruction set is designed to support the large register file. It includes a full complement of 8-bit arithmetic and logic operations. There are 41 instructions. No special I/O instructions are necessary because I/O control and data registers are mapped directly into the register file. Flexible instructions for bit addressing, rotate, and shift operations complete the powerful data manipulation capabilities of the SAM88RCRI instruction set.

REGISTER ADDRESSING

To access an individual register, an 8-bit address in the range 0-255 or the 4-bit address of a working register is specified. Paired registers can be used to construct 13-bit program memory or data memory addresses. For detailed information about register addressing, please refer to Chapter 2, "Address Spaces".

ADDRESSING MODES

There are six addressing modes: Register (R), Indirect Register (IR), Indexed (X), Direct (DA), Relative (RA), and Immediate (IM). For detailed descriptions of these addressing modes, please refer to Chapter 3, "Addressing Modes".

Table 6-1. Instruction Group Summary

Mnemonic	Operands	Instruction
Load Instructions		
CLR	dst	Clear
LD	dst,src	Load
LDC	dst,src	Load program memory
LDE	dst,src	Load external data memory
LDCD	dst,src	Load program memory and decrement
LDED	dst,src	Load external data memory and decrement
LDCI	dst,src	Load program memory and increment
LDEI	dst,src	Load external data memory and increment
POP	dst	Pop from stack
PUSH	src	Push to stack
Arithmetic Instructions		
ADC	dst,src	Add with carry
ADD	dst,src	Add
CP	dst,src	Compare
DEC	dst	Decrement
INC	dst	Increment
SBC	dst,src	Subtract with carry
SUB	dst,src	Subtract
Logic Instructions		
AND	dst,src	Logical AND
COM	dst	Complement
OR	dst,src	Logical OR
XOR	dst,src	Logical exclusive OR

Table 6-1. Instruction Group Summary (Continued)

Mnemonic	Operands	Instruction
Program Control Instructions		
CALL	dst	Call procedure
IRET		Interrupt return
JP	cc,dst	Jump on condition code
JP	dst	Jump unconditional
JR	cc,dst	Jump relative on condition code
RET		Return
Bit Manipulation Instructions		
TCM	dst,src	Test complement under mask
TM	dst,src	Test under mask
Rotate and Shift Instructions		
RL	dst	Rotate left
RLC	dst	Rotate left through carry
RR	dst	Rotate right
RRC	dst	Rotate right through carry
SRA	dst	Shift right arithmetic
CPU Control Instructions		
CCF		Complement carry flag
DI		Disable interrupts
EI		Enable interrupts
IDLE		Enter Idle mode
NOP		No operation
RCF		Reset carry flag
SCF		Set carry flag
STOP		Enter stop mode

FLAGS REGISTER (FLAGS)

The flags register FLAGS contains eight bits that describe the current status of CPU operations. Four of these bits, FLAGS.4–FLAGS.7, can be tested and used with conditional jump instructions;

FLAGS register can be set or reset by instructions as long as its outcome does not affect the flags, such as, Load instruction. Logical and Arithmetic instructions such as, AND, OR, XOR, ADD, and SUB can affect the Flags register. For example, the AND instruction updates the Zero, Sign and Overflow flags based on the outcome of the AND instruction. If the AND instruction uses the Flags register as the destination, then simultaneously, two write will occur to the Flags register producing an unpredictable result.

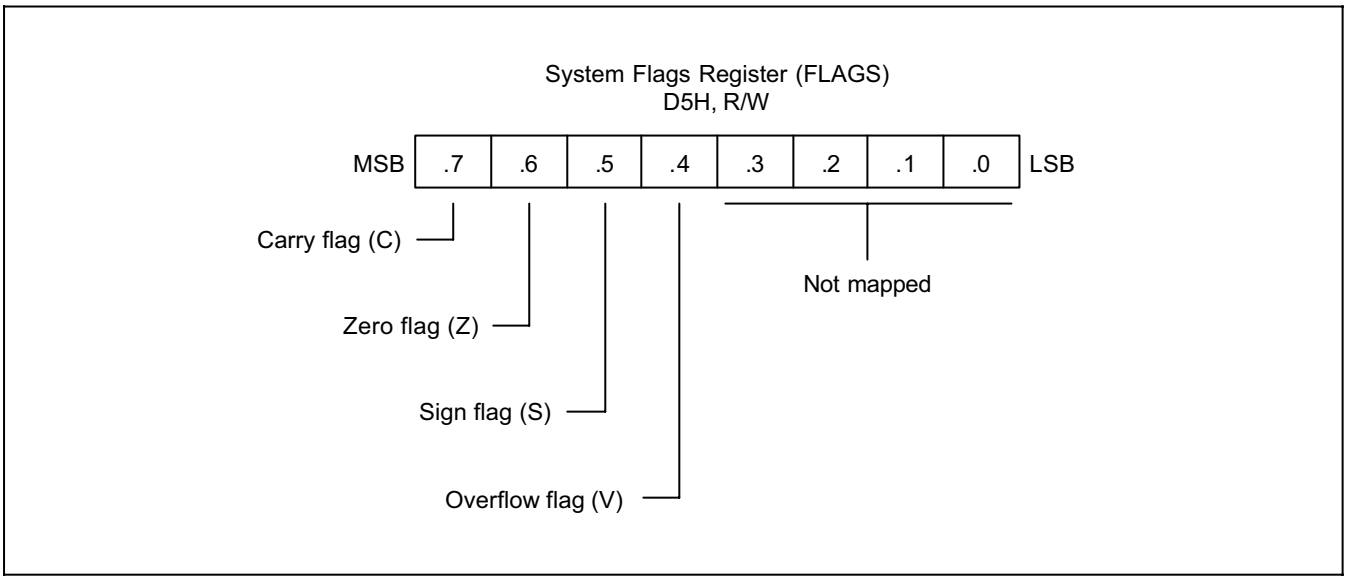


Figure 6-1. System Flags Register (FLAGS)

FLAG DESCRIPTIONS

Overflow Flag (FLAGS.4, V)

The V flag is set to "1" when the result of a two's-complement operation is greater than + 127 or less than – 128. It is also cleared to "0" following logic operations.

Sign Flag (FLAGS.5, S)

Following arithmetic, logic, rotate, or shift operations, the sign bit identifies the state of the MSB of the result. A logic zero indicates a positive number and a logic one indicates a negative number.

Zero Flag (FLAGS.6, Z)

For arithmetic and logic operations, the Z flag is set to "1" if the result of the operation is zero. For operations that test register bits, and for shift and rotate operations, the Z flag is set to "1" if the result is logic zero.

Carry Flag (FLAGS.7, C)

The C flag is set to "1" if the result from an arithmetic operation generates a carry-out from or a borrow to the bit 7 position (MSB). After rotate and shift operations, it contains the last value shifted out of the specified register. Program instructions can set, clear, or complement the carry flag.

INSTRUCTION SET NOTATION

Table 6-2. Flag Notation Conventions

Flag	Description
C	Carry flag
Z	Zero flag
S	Sign flag
V	Overflow flag
0	Cleared to logic zero
1	Set to logic one
*	Set or cleared according to operation
–	Value is unaffected
x	Value is undefined

Table 6-3. Instruction Set Symbols

Symbol	Description
dst	Destination operand
src	Source operand
@	Indirect register address prefix
PC	Program counter
FLAGS	Flags register (D5H)
#	Immediate operand or register address prefix
H	Hexadecimal number suffix
D	Decimal number suffix
B	Binary number suffix
opc	Opcode

Table 6-4. Instruction Notation Conventions

Notation	Description	Actual Operand Range
cc	Condition code	See list of condition codes in Table 6-6.
r	Working register only	Rn (n = 0–15)
rr	Working register pair	RRp (p = 0, 2, 4, ..., 14)
R	Register or working register	reg or Rn (reg = 0–255, n = 0–15)
RR	Register pair or working register pair	reg or RRp (reg = 0–254, even number only, where p = 0, 2, ..., 14)
Ir	Indirect working register only	@Rn (n = 0–15)
IR	Indirect register or indirect working register	@Rn or @reg (reg = 0–255, n = 0–15)
Irr	Indirect working register pair only	@RRp (p = 0, 2, ..., 14)
IRR	Indirect register pair or indirect working register pair	@RRp or @reg (reg = 0–254, even only, where p = 0, 2, ..., 14)
X	Indexed addressing mode	#reg[Rn] (reg = 0–255, n = 0–15)
XS	Indexed (short offset) addressing mode	#addr[RRp] (addr = range – 128 to + 127, where p = 0, 2, ..., 14)
XL	Indexed (long offset) addressing mode	#addr [RRp] (addr = range 0–8191, where p = 0, 2, ..., 14)
DA	Direct addressing mode	addr (addr = range 0–8191)
RA	Relative addressing mode	addr (addr = number in the range + 127 to – 128 that is an offset relative to the address of the next instruction)
IM	Immediate addressing mode	#data (data = 0–255)

Table 6-5. Opcode Quick Reference

OPCODE MAP									
	LOWER NIBBLE (HEX)								
	–	0	1	2	3	4	5	6	7
U	0	DEC R1	DEC IR1	ADD r1,r2	ADD r1,Ir2	ADD R2,R1	ADD IR2,R1	ADD R1,IM	
P	1	RLC R1	RLC IR1	ADC r1,r2	ADC r1,Ir2	ADC R2,R1	ADC IR2,R1	ADC R1,IM	
P	2	INC R1	INC IR1	SUB r1,r2	SUB r1,Ir2	SUB R2,R1	SUB IR2,R1	SUB R1,IM	
E	3	JP IRR1		SBC r1,r2	SBC r1,Ir2	SBC R2,R1	SBC IR2,R1	SBC R1,IM	
R	4			OR r1,r2	OR r1,Ir2	OR R2,R1	OR IR2,R1	OR R1,IM	
	5	POP R1	POP IR1	AND r1,r2	AND r1,Ir2	AND R2,R1	AND IR2,R1	AND R1,IM	
N	6	COM R1	COM IR1	TCM r1,r2	TCM r1,Ir2	TCM R2,R1	TCM IR2,R1	TCM R1,IM	
I	7	PUSH R2	PUSH IR2	TM r1,r2	TM r1,Ir2	TM R2,R1	TM IR2,R1	TM R1,IM	
B	8								LD r1, x, r2
B	9	RL R1	RL IR1						LD r2, x, r1
L	A			CP r1,r2	CP r1,Ir2	CP R2,R1	CP IR2,R1	CP R1,IM	LDC r1, Irr2, xL
E	B	CLR R1	CLR IR1	XOR r1,r2	XOR r1,Ir2	XOR R2,R1	XOR IR2,R1	XOR R1,IM	LDC r2, Irr2, xL
	C	RRC R1	RRC IR1		LDC r1,Irr2				LD r1, Ir2
H	D	SRA R1	SRA IR1		LDC r2,Irr1			LD IR1,IM	LD Ir1, r2
E	E	RR R1	RR IR1	LDCD r1,Irr2	LDCI r1,Irr2	LD R2,R1	LD R2,IR1	LD R1,IM	LDC r1, Irr2, xs
X	F					CALL IRR1	LD IR2,R1	CALL DA1	LDC r2, Irr1, xs

Table 6-5. Opcode Quick Reference (Continued)

OPCODE MAP									
LOWER NIBBLE (HEX)									
	–	8	9	A	B	C	D	E	F
U	0	LD r1,R2	LD r2,R1		JR cc,RA	LD r1,IM	JP cc,DA	INC r1	
P	1	–	–		–	–	–	–	
P	2								
E	3								
R	4								
	5								
N	6								IDLE
I	7	–	–		–	–	–	–	STOP
B	8								DI
B	9								EI
L	A								RET
E	B								IRET
	C								RCF
H	D	–	–		–	–	–	–	SCF
E	E								CCF
X	F	LD r1,R2	LD r2,R1		JR cc,RA	LD r1,IM	JP cc,DA	INC r1	NOP

SAMSUNG ELECTRONICS

CONDITION CODES

The opcode of a conditional jump always contains a 4-bit field called the condition code (cc). This specifies under which conditions it is to execute the jump. For example, a conditional jump with the condition code for "equal" after a compare operation only jumps if the two operands are equal. Condition codes are listed in Table 6-6.

The carry (C), zero (Z), sign (S), and overflow (V) flags are used to control the operation of conditional jump instructions.

Table 6-6. Condition Codes

Binary	Mnemonic	Description	Flags Set
0000	F	Always false	–
1000	T	Always true	–
0111 (1)	C	Carry	C = 1
1111 (1)	NC	No carry	C = 0
0110 (1)	Z	Zero	Z = 1
1110 (1)	NZ	Not zero	Z = 0
1101	PL	Plus	S = 0
0101	MI	Minus	S = 1
0100	OV	Overflow	V = 1
1100	NOV	No overflow	V = 0
0110 (1)	EQ	Equal	Z = 1
1110 (1)	NE	Not equal	Z = 0
1001	GE	Greater than or equal	(S XOR V) = 0
0001	LT	Less than	(S XOR V) = 1
1010	GT	Greater than	(Z OR (S XOR V)) = 0
0010	LE	Less than or equal	(Z OR (S XOR V)) = 1
1111 (1)	UGE	Unsigned greater than or equal	C = 0
0111 (1)	ULT	Unsigned less than	C = 1
1011	UGT	Unsigned greater than	(C = 0 AND Z = 0) = 1
0011	ULE	Unsigned less than or equal	(C OR Z) = 1

NOTES:

1. It indicates condition codes that are related to two different mnemonics but which test the same flag. For example, Z and EQ are both true if the zero flag (Z) is set, but after an ADD instruction, Z would probably be used; after a CP instruction, however, EQ would probably be used.
2. For operations involving unsigned numbers, the special condition codes UGE, ULT, UGT, and ULE must be used.

INSTRUCTION DESCRIPTIONS

This section contains detailed information and programming examples for each instruction in the SAM88RCRI instruction set. Information is arranged in a consistent format for improved readability and for fast referencing. The following information is included in each instruction description:

— Instruction name (mnemonic)
— Full instruction name
— Source/destination format of the instruction operand
— Shorthand notation of the instruction's operation
— Textual description of the instruction's effect
— Specific flag settings affected by the instruction
— Detailed description of the instruction's format, execution time, and addressing mode(s)
— Programming example(s) explaining how to use the instruction

ADC — Add with Carry

ADC dst,src

Operation: dst _ dst + src + c

The source operand, along with the setting of the carry flag, is added to the destination operand and the sum is stored in the destination. The contents of the source are unaffected.
Two's-complement addition is performed. In multiple precision arithmetic, this instruction permits the carry from the addition of low-order operands to be carried into the addition of high-order operands.

Flags:
C: Set if there is a carry from the most significant bit of the result; cleared otherwise.
Z: Set if the result is "0"; cleared otherwise.
S: Set if the result is negative; cleared otherwise.
V: Set if arithmetic overflow occurs, that is, if both operands are of the same sign and the result is of the opposite sign; cleared otherwise.

Format:

	Bytes	Cycles	Opcode (Hex)	Addr Mode dst	Addr Mode src
opc | dst | src	2	4	12	r	r
		6	13	r	lr
opc | src | dst	3	6	14	R	R
		6	15	R	IR
opc | dst | src	3	6	16	R	IM

Examples: Given: R1 = 10H, R2 = 03H, C flag = "1", register 01H = 20H, register 02H = 03H, and register 03H = 0AH:

ADC R1,R2 ® R1 = 14H, R2 = 03H
ADC R1,@R2 ® R1 = 1BH, R2 = 03H
ADC 01H,02H ® Register 01H = 24H, register 02H = 03H
ADC 01H,@02H ® Register 01H = 2BH, register 02H = 03H
ADC 01H,#11H ® Register 01H = 32H

In the first example, destination register R1 contains the value 10H, the carry flag is set to "1", and the source working register R2 contains the value 03H. The statement "ADC R1,R2" adds 03H and the carry flag value ("1") to the destination value 10H, leaving 14H in register R1.

ADD — Add

ADD dst,src

Operation: dst _ dst + src

The source operand is added to the destination operand and the sum is stored in the destination. The contents of the source are unaffected. Two's-complement addition is performed.

Flags:

C: Set if there is a carry from the most significant bit of the result; cleared otherwise.
Z: Set if the result is "0"; cleared otherwise.
S: Set if the result is negative; cleared otherwise.
V: Set if arithmetic overflow occurred, that is, if both operands are of the same sign and the result is of the opposite sign; cleared otherwise.

Format:

	Bytes	Cycles	Opcode (Hex)	Addr Mode dst	src
opc | dst | src	2	4	02	r	r
		6	03	r	lr
opc | src | dst	3	6	04	R	R
		6	05	R	IR
opc | dst | src	3	6	06	R	IM

Examples: Given: R1 = 12H, R2 = 03H, register 01H = 21H, register 02H = 03H, register 03H = 0AH:

ADD	R1,R2	®	R1 = 15H, R2 = 03H
ADD	R1,@R2	®	R1 = 1CH, R2 = 03H
ADD	01H,02H	®	Register 01H = 24H, register 02H = 03H
ADD	01H,@02H	®	Register 01H = 2BH, register 02H = 03H
ADD	01H,#25H	®	Register 01H = 46H

In the first example, destination working register R1 contains 12H and the source working register R2 contains 03H. The statement "ADD R1,R2" adds 03H to 12H, leaving the value 15H in register R1.

AND — Logical AND

AND dst,src

Operation: dst _ dst AND src

The source operand is logically ANDed with the destination operand. The result is stored in the destination. The AND operation results in a "1" bit being stored whenever the corresponding bits in the two operands are both logic ones; otherwise a "0" bit value is stored. The contents of the source are unaffected.

Flags:
C: Unaffected.
Z: Set if the result is "0"; cleared otherwise.
S: Set if the result bit 7 is set; cleared otherwise.
V: Always cleared to "0".

Format:

	Bytes	Cycles	Opcode (Hex)	Addr Mode dst	src
opc | dst | src	2	4	52	r	r
		6	53	r	lr
opc | src | dst	3	6	54	R	R
		6	55	R	IR
opc | dst | src	3	6	56	R	IM

Examples: Given: R1 = 12H, R2 = 03H, register 01H = 21H, register 02H = 03H, register 03H = 0AH:

AND R1,R2 ® R1 = 02H, R2 = 03H
AND R1,@R2 ® R1 = 02H, R2 = 03H
AND 01H,02H ® Register 01H = 01H, register 02H = 03H
AND 01H,@02H ® Register 01H = 00H, register 02H = 03H
AND 01H,#25H ® Register 01H = 21H

In the first example, destination working register R1 contains the value 12H and the source working register R2 contains 03H. The statement "AND R1,R2" logically ANDs the source operand 03H with the destination operand value 12H, leaving the value 02H in register R1.

CALL — Call Procedure

CALL dst

Operation:

SP ← SP – 1
@SP ← PCL
SP ← SP –1
@SP ← PCH
PC ← dst

The current contents of the program counter are pushed onto the top of the stack. The program counter value used is the address of the first instruction following the CALL instruction. The specified destination address is then loaded into the program counter and points to the first instruction of a procedure. At the end of the procedure the return instruction (RET) can be used to return to the original program flow. RET pops the top of the stack back into the program counter.

Flags: No flags are affected.

Format:

	Bytes	Cycles	Opcode (Hex)	Addr Mode dst
opc | dst	3	14	F6	DA
opc | dst	2	12	F4	IRR

Examples: Given: R0 = 15H, R1 = 21H, PC = 1A47H, and SP = 0B2H:

CALL 1521H ® SP = 0B0H
(Memory locations 00H = 1AH, 01H = 4AH, where 4AH is the address that follows the instruction.)

CALL @RR0 ® SP = 0B0H (00H = 1AH, 01H = 49H)

In the first example, if the program counter value is 1A47H and the stack pointer contains the value 0B2H, the statement "CALL 1521H" pushes the current PC value onto the top of the stack. The stack pointer now points to memory location 00H. The PC is then loaded with the value 1521H, the address of the first instruction in the program sequence to be executed.

If the contents of the program counter and stack pointer are the same as in the first example, the statement "CALL @RR0" produces the same result except that the 49H is stored in stack location 01H (because the two-byte instruction format was used). The PC is then loaded with the value 1521H, the address of the first instruction in the program sequence to be executed.

CCF — Complement Carry Flag

CCF

Operation: C _ NOT C

The carry flag (C) is complemented. If C = "1", the value of the carry flag is changed to logic zero; if C = "0", the value of the carry flag is changed to logic one.

Flags: **C:** Complemented.

No other flags are affected.

Format:

	Bytes	Cycles	Opcode (Hex)
opc	1	4	EF

Example: Given: The carry flag = "0":

CCF

If the carry flag = "0", the CCF instruction complements it in the FLAGS register (0D5H), changing its value from logic zero to logic one.

CLR — Clear

CLR dst

Operation: dst _ "0"

The destination location is cleared to "0".

Flags: No flags are affected.

Format:

	Bytes	Cycles	Opcode (Hex)	Addr Mode dst
opc | dst	2	4	B0	R
		4	B1	IR

Examples: Given: Register 00H = 4FH, register 01H = 02H, and register 02H = 5EH:

CLR 00H ® Register 00H = 00H

CLR @01H ® Register 01H = 02H, register 02H = 00H

In Register (R) addressing mode, the statement "CLR 00H" clears the destination register 00H value to 00H. In the second example, the statement "CLR @01H" uses Indirect Register (IR) addressing mode to clear the 02H register value to 00H.

COM — Complement

COM dst

Operation: dst _ NOT dst

The contents of the destination location are complemented (one's complement); all "1s" are changed to "0s", and vice-versa.

Flags:
C: Unaffected.
Z: Set if the result is "0"; cleared otherwise.
S: Set if the result bit 7 is set; cleared otherwise.
V: Always reset to "0".

Format:

	Bytes	Cycles	Opcode (Hex)	Addr Mode dst
opc | dst	2	4	60	R
		4	61	IR

Examples: Given: R1 = 07H and register 07H = 0F1H:

COM R1 ® R1 = 0F8H

COM @R1 ® R1 = 07H, register 07H = 0EH

In the first example, destination working register R1 contains the value 07H (00000111B). The statement "COM R1" complements all the bits in R1: all logic ones are changed to logic zeros, and vice-versa, leaving the value 0F8H (11111000B).

In the second example, Indirect Register (IR) addressing mode is used to complement the value of destination register 07H (11110001B), leaving the new value 0EH (00001110B).

CP — Compare

CP dst,src

Operation: dst – src

The source operand is compared to (subtracted from) the destination operand, and the appropriate flags are set accordingly. The contents of both operands are unaffected by the comparison.

Flags:
C: Set if a "borrow" occurred (src > dst); cleared otherwise.
Z: Set if the result is "0"; cleared otherwise.
S: Set if the result is negative; cleared otherwise.
V: Set if arithmetic overflow occurred, that is, if the operands were of opposite signs and the sign of the result is of the same as the sign of the source operand; cleared otherwise.

Format:

	Bytes	Cycles	Opcode (Hex)	Addr Mode dst	src
opc | dst | src	2	4	A2	r	r
		6	A3	r	lr
opc | src | dst	3	6	A4	R	R
		6	A5	R	IR
opc | dst | src	3	6	A6	R	IM

Examples: 1. Given: R1 = 02H and R2 = 03H:

```
CP      R1,R2   →   Set the C and S flags
```

Destination working register R1 contains the value 02H and source register R2 contains the value 03H. The statement "CP R1,R2" subtracts the R2 value (source/subtrahend) from the R1 value (destination/minuend). Because a "borrow" occurs and the difference is negative, C and S are "1".

2. Given: R1 = 05H and R2 = 0AH:

```
        CP      R1,R2
        JP      UGE,SKIP
        INC     R1
SKIP    LD      R3,R1
```

In this example, destination working register R1 contains the value 05H which is less than the contents of the source working register R2 (0AH). The statement "CP R1,R2" generates C = "1" and the JP instruction does not jump to the SKIP location. After the statement "LD R3,R1" executes, the value 06H remains in working register R3.

DEC — Decrement

DEC dst

Operation: dst _ dst – 1

The contents of the destination operand are decremented by one.

Flags:
C: Unaffected.
Z: Set if the result is "0"; cleared otherwise.
S: Set if result is negative; cleared otherwise.
V: Set if arithmetic overflow occurred, that is, dst value is – 128 (80H) and result value is + 127 (7FH); cleared otherwise.

Format:

	Bytes	Cycles	Opcode (Hex)	Addr Mode dst
opc | dst	2	4	00	R
		4	01	IR

Examples: Given: R1 = 03H and register 03H = 10H:

DEC R1 ® R1 = 02H
DEC @R1 ® Register 03H = 0FH

In the first example, if working register R1 contains the value 03H, the statement "DEC R1" decrements the hexadecimal value by one, leaving the value 02H. In the second example, the statement "DEC @R1" decrements the value 10H contained in the destination register 03H by one, leaving the value 0FH.

DI — Disable Interrupts

DI

Operation: SYM (3) _ 0

Bit zero of the system mode register, SYM.3, is cleared to "0", globally disabling all interrupt processing. Interrupt requests will continue to set their respective interrupt pending bits, but the CPU will not service them while interrupt processing is disabled.

Flags: No flags are affected.

Format:

	Bytes	Cycles	Opcode (Hex)
opc	1	4	8F

Example: Given: SYM = 08H:

DI

If the value of the SYM register is 08H, the statement "DI" leaves the new value 00H in the register and clears SYM.3 to "0", disabling interrupt processing.

EI — Enable Interrupts

EI

Operation: SYM (3) _ 1

An EI instruction sets bit 3 of the system mode register, SYM.3 to "1". This allows interrupts to be serviced as they occur. If an interrupt's pending bit was set while interrupt processing was disabled (by executing a DI instruction), it will be serviced when you execute the EI instruction.

Flags: No flags are affected.

Format:

	Bytes	Cycles	Opcode (Hex)
opc	1	4	9F

Example: Given: SYM = 00H:

EI

If the SYM register contains the value 00H, that is, if interrupts are currently disabled, the statement "EI" sets the SYM register to 08H, enabling all interrupts. (SYM.3 is the enable bit for global interrupt processing.)

IDLE — Idle Operation

IDLE

Operation:

The IDLE instruction stops the CPU clock while allowing system clock oscillation to continue. Idle mode can be released by an interrupt request (IRQ) or an external reset operation.

Flags: No flags are affected.

Format:

	Bytes	Cycles	Opcode (Hex)	Addr Mode dst	src
opc	1	4	6F	–	–

Example: The instruction

```
IDLE
NOP
NOP
NOP
```

stops the CPU clock but not the system clock.

INC — Increment

INC dst

Operation: dst _ dst + 1

The contents of the destination operand are incremented by one.

Flags:
C: Unaffected.
Z: Set if the result is "0"; cleared otherwise.
S: Set if the result is negative; cleared otherwise.
V: Set if arithmetic overflow occurred, that is dst value is + 127 (7FH) and result is – 128 (80H); cleared otherwise.

Format:

	Bytes	Cycles	Opcode (Hex)	Addr Mode dst
dst | opc	1	4	rE	r
			r = 0 to F	
opc | dst	2	4	20	R
		4	21	IR

Examples: Given: R0 = 1BH, register 00H = 0CH, and register 1BH = 0FH:

INC R0 ® R0 = 1CH
INC 00H ® Register 00H = 0DH
INC @R0 ® R0 = 1BH, register 01H = 10H

In the first example, if destination working register R0 contains the value 1BH, the statement "INC R0" leaves the value 1CH in that same register.

The next example shows the effect an INC instruction has on register 00H, assuming that it contains the value 0CH.

In the third example, INC is used in Indirect Register (IR) addressing mode to increment the value of register 1BH from 0FH to 10H.

IRET — Interrupt Return

IRET IRET

Operation:
FLAGS _ @SP
SP _ SP + 1
PC _ @SP
SP _ SP + 2
SYM(2) _ 1

This instruction is used at the end of an interrupt service routine. It restores the flag register and the program counter. It also re-enables global interrupts.

Flags: All flags are restored to their original settings (that is, the settings before the interrupt occurred).

Format:

IRET (Normal)	Bytes	Cycles	Opcode (Hex)
opc	1	10	BF
		12	

JP — Jump

JP cc,dst (Conditional)

JP dst (Unconditional)

Operation: If cc is true, PC _ dst

The conditional JUMP instruction transfers program control to the destination address if the condition specified by the condition code (cc) is true; otherwise, the instruction following the JP instruction is executed. The unconditional JP simply replaces the contents of the PC with the contents of the specified register pair. Control then passes to the statement addressed by the PC.

Flags: No flags are affected.

Format: (1)

(2)	Bytes	Cycles	Opcode (Hex)	Addr Mode dst
cc | opc / dst	3	8	ccD cc = 0 to F	DA
opc / dst	2	8	30	IRR

NOTES:
1. The 3-byte format is used for a conditional jump and the 2-byte format for an unconditional jump.
2. In the first byte of the three-byte instruction format (conditional jump), the condition code and the op code are both four bits.

Examples: Given: The carry flag (C) = "1", register 00 = 01H, and register 01 = 20H:

JP C,LABEL_W ® LABEL_W = 1000H, PC = 1000H

JP @00H ® PC = 0120H

The first example shows a conditional JP. Assuming that the carry flag is set to "1", the statement "JP C,LABEL_W" replaces the contents of the PC with the value 1000H and transfers control to that location. Had the carry flag not been set, control would then have passed to the statement immediately following the JP instruction.

The second example shows an unconditional JP. The statement "JP @00" replaces the contents of the PC with the contents of the register pair 00H and 01H, leaving the value 0120H.

JR — Jump Relative

JR cc,dst

Operation: If cc is true, PC _ PC + dst

If the condition specified by the condition code (cc) is true, the relative address is added to the program counter and control passes to the statement whose address is now in the program counter; otherwise, the instruction following the JR instruction is executed (See list of condition codes).

The range of the relative address is + 127, – 128, and the original value of the program counter is taken to be the address of the first instruction byte following the JR statement.

Flags: No flags are affected.

Format:

(note)	Bytes	Cycles	Opcode (Hex)	Addr Mode dst
cc | opc | dst	2	6	ccB	RA
			cc = 0 to F	

NOTE: In the first byte of the two-byte instruction format, the condition code and the op code are each four bits.

Example: Given: The carry flag = "1" and LABEL_X = 1FF7H:

JR C,LABEL_X ® PC = 1FF7H

If the carry flag is set (that is, if the condition code is true), the statement "JR C,LABEL_X" will pass control to the statement whose address is now in the PC. Otherwise, the program instruction following the JR would be executed.

LD — Load

LD dst,src

Operation: dst _ src

The contents of the source are loaded into the destination. The source's contents are unaffected.

Flags: No flags are affected.

Format:

			Bytes	Cycles	Opcode (Hex)	Addr Mode dst	src
dst | opc	src		2	4	rC	r	IM
				4	r8	r	R
src | opc	dst		2	4	r9	R	r
					r = 0 to F		
opc	dst | src		2	4	C7	r	Ir
				4	D7	Ir	r
opc	src	dst	3	6	E4	R	R
				6	E5	R	IR
opc	dst	src	3	6	E6	R	IM
				6	D6	IR	IM
opc	src	dst	3	6	F5	IR	R
opc	dst | src	x	3	6	87	r	x [r]
opc	src | dst	x	3	6	97	x [r]	r

LD — Load

LD (Continued)

Examples: Given: R0 = 01H, R1 = 0AH, register 00H = 01H, register 01H = 20H, register 02H = 02H, LOOP = 30H, and register 3AH = 0FFH:

LD	R0,#10H	®	R0 = 10H
LD	R0,01H	®	R0 = 20H, register 01H = 20H
LD	01H,R0	®	Register 01H = 01H, R0 = 01H
LD	R1,@R0	®	R1 = 20H, R0 = 01H
LD	@R0,R1	®	R0 = 01H, R1 = 0AH, register 01H = 0AH
LD	00H,01H	®	Register 00H = 20H, register 01H = 20H
LD	02H,@00H	®	Register 02H = 20H, register 00H = 01H
LD	00H,#0AH	®	Register 00H = 0AH
LD	@00H,#10H	®	Register 00H = 01H, register 01H = 10H
LD	@00H,02H	®	Register 00H = 01H, register 01H = 02, register 02H = 02H
LD	R0,#LOOP[R1]	®	R0 = 0FFH, R1 = 0AH
LD	#LOOP[R0],R1	®	Register 31H = 0AH, R0 = 01H, R1 = 0AH

LDC/LDE — Load Memory

LDC/LDE dst,src

Operation: dst _ src

This instruction loads a byte from program or data memory into a working register or vice-versa. The source values are unaffected. LDC refers to program memory and LDE to data memory. The assembler makes "Irr" or "rr" values an even number for program memory and odd an odd number for data memory.

Flags: No flags are affected.

Format:

					Bytes	Cycles	Opcode (Hex)	Addr Mode dst	Addr Mode src
1.	opc	dst | src			2	10	C3	r	Irr
2.	opc	src | dst			2	10	D3	Irr	r
3.	opc	dst | src	XS		3	12	E7	r	XS [rr]
4.	opc	src | dst	XS		3	12	F7	XS [rr]	r
5.	opc	dst | src	XL_L	XL_H	4	14	A7	r	XL [rr]
6.	opc	src | dst	XL_L	XL_H	4	14	B7	XL [rr]	r
7.	opc	dst | 0000	DA_L	DA_H	4	14	A7	r	DA
8.	opc	src | 0000	DA_L	DA_H	4	14	B7	DA	r
9.	opc	dst | 0001	DA_L	DA_H	4	14	A7	r	DA
10.	opc	src | 0001	DA_L	DA_H	4	14	B7	DA	r

NOTES:

1. The source (src) or working register pair [rr] for formats 5 and 6 cannot use register pair 0–1.
2. For formats 3 and 4, the destination address "XS [rr]" and the source address "XS [rr]" are each one byte.
3. For formats 5 and 6, the destination address "XL [rr]" and the source address "XL [rr]" are each two bytes.
4. The DA and r source values for formats 7 and 8 are used to address program memory; the second set of values, used in formats 9 and 10, are used to address data memory.

LDC/LDE — Load Memory

LDC/LDE (Continued)

Examples: Given: R0 = 11H, R1 = 34H, R2 = 01H, R3 = 04H, R4 = 00H, R5 = 60H; Program memory locations 0061 = AAH, 0103H = 4FH, 0104H = 1A, 0105H = 6DH, and 1104H = 88H. External data memory locations 0061H = BBH, 0103H = 5FH, 0104H = 2AH, 0105H = 7DH, and 1104H = 98H:

```
LDC     R0,@RR2          ;  R0 _ contents of program memory location 0104H
                         ;  R0 = 1AH, R2 = 01H, R3 = 04H

LDE     R0,@RR2          ;  R0 _ contents of external data memory location 0104H
                         ;  R0 = 2AH, R2 = 01H, R3 = 04H

LDC (note) @RR2,R0       ;  11H (contents of R0) is loaded into program memory
                         ;  location 0104H (RR2),
                         ;  working registers R0, R2, R3 _ no change

LDE     @RR2,R0          ;  11H (contents of R0) is loaded into external data memory
                         ;  location 0104H (RR2),
                         ;  working registers R0, R2, R3 _ no change

LDC     R0,#01H[RR4]     ;  R0 _ contents of program memory location 0061H
                         ;  (01H + RR4),
                         ;  R0 = AAH, R2 = 00H, R3 = 60H

LDE     R0,#01H[RR4]     ;  R0 _ contents of external data memory location 0061H
                         ;  (01H + RR4), R0 = BBH, R4 = 00H, R5 = 60H

LDC (note) #01H[RR4],R0  ;  11H (contents of R0) is loaded into program memory location
                         ;  0061H (01H + 0060H)

LDE     #01H[RR4],R0     ;  11H (contents of R0) is loaded into external data memory
                         ;  location 0061H (01H + 0060H)

LDC     R0,#1000H[RR2]   ;  R0 _ contents of program memory location 1104H
                         ;  (1000H + 0104H), R0 = 88H, R2 = 01H, R3 = 04H

LDE     R0,#1000H[RR2]   ;  R0 _ contents of external data memory location 1104H
                         ;  (1000H + 0104H), R0 = 98H, R2 = 01H, R3 = 04H

LDC     R0,1104H         ;  R0 _ contents of program memory location 1104H, R0 = 88H

LDE     R0,1104H         ;  R0 _ contents of external data memory location 1104H,
                         ;  R0 = 98H

LDC (note) 1105H,R0      ;  11H (contents of R0) is loaded into program memory location
                         ;  1105H, (1105H) _ 11H

LDE     1105H,R0         ;  11H (contents of R0) is loaded into external data memory
                         ;  location 1105H, (1105H) _ 11H
```

NOTE: These instructions are not supported by masked ROM type devices.

LDCD/LDED — Load Memory and Decrement

LDCD/LDED dst,src

Operation: dst _ src

rr _ rr – 1

These instructions are used for user stacks or block transfers of data from program or data memory to the register file. The address of the memory location is specified by a working register pair. The contents of the source location are loaded into the destination location. The memory address is then decremented. The contents of the source are unaffected.

LDCD references program memory and LDED references external data memory. The assembler makes "lrr" an even number for program memory and an odd number for data memory.

Flags: No flags are affected.

Format:

	Bytes	Cycles	Opcode (Hex)	Addr Mode dst	src
opc | dst | src	2	10	E2	r	lrr

Examples: Given: R6 = 10H, R7 = 33H, R8 = 12H, program memory location 1033H = 0CDH, and external data memory location 1033H = 0DDH:

```
LDCD    R8,@RR6     ;  0CDH (contents of program memory location 1033H) is loaded
                    ;  into R8 and RR6 is decremented by one
                    ;  R8 = 0CDH, R6 = 10H, R7 = 32H (RR6 _ RR6 – 1)

LDED    R8,@RR6     ;  0DDH (contents of data memory location 1033H) is loaded
                    ;  into R8 and RR6 is decremented by one (RR6 _ RR6 – 1)
                    ;  R8 = 0DDH, R6 = 10H, R7 = 32H
```

LDCI/LDEI — LOAD MEMORY AND INCREMENT

LDCI/LDEI dst,src

Operation: dst _ src

rr _ rr + 1

These instructions are used for user stacks or block transfers of data from program or data memory to the register file. The address of the memory location is specified by a working register pair. The contents of the source location are loaded into the destination location. The memory address is then incremented automatically. The contents of the source are unaffected.

LDCI refers to program memory and LDEI refers to external data memory. The assembler makes "lrr" even for program memory and odd for data memory.

Flags: No flags are affected.

Format:

	Bytes	Cycles	Opcode (Hex)	Addr Mode dst	src
opc | dst | src	2	10	E3	r	lrr

Examples: Given: R6 = 10H, R7 = 33H, R8 = 12H, program memory locations 1033H = 0CDH and 1034H = 0C5H; external data memory locations 1033H = 0DDH and 1034H = 0D5H:

```
LDCI    R8,@RR6     ;  0CDH (contents of program memory location 1033H) is loaded
                    ;  into R8 and RR6 is incremented by one (RR6 _ RR6 + 1)
                    ;  R8 = 0CDH, R6 = 10H, R7 = 34H

LDEI    R8,@RR6     ;  0DDH (contents of data memory location 1033H) is loaded
                    ;  into R8 and RR6 is incremented by one (RR6 _ RR6 + 1)
                    ;  R8 = 0DDH, R6 = 10H, R7 = 34H
```

NOP — No Operation

NOP

Operation: No action is performed when the CPU executes this instruction. Typically, one or more NOPs are executed in sequence in order to effect a timing delay of variable duration.

Flags: No flags are affected.

Format:

	Bytes	Cycles	Opcode (Hex)
opc	1	4	FF

Example: When the instruction

NOP

is encountered in a program, no operation occurs. Instead, there is a delay in instruction execution time.

OR — Logical OR

OR dst,src

Operation: dst _ dst OR src

The source operand is logically ORed with the destination operand and the result is stored in the destination. The contents of the source are unaffected. The OR operation results in a "1" being stored whenever either of the corresponding bits in the two operands is a "1"; otherwise a "0" is stored.

Flags:
C: Unaffected.
Z: Set if the result is "0"; cleared otherwise.
S: Set if the result bit 7 is set; cleared otherwise.
V: Always cleared to "0".

Format:

	Bytes	Cycles	Opcode (Hex)	Addr Mode dst	src
opc | dst | src	2	4	42	r	r
		6	43	r	lr
opc | src | dst	3	6	44	R	R
		6	45	R	IR
opc | dst | src	3	6	46	R	IM

Examples: Given: R0 = 15H, R1 = 2AH, R2 = 01H, register 00H = 08H, register 01H = 37H, and register 08H = 8AH:

OR R0,R1 ® R0 = 3FH, R1 = 2AH
OR R0,@R2 ® R0 = 37H, R2 = 01H, register 01H = 37H
OR 00H,01H ® Register 00H = 3FH, register 01H = 37H
OR 01H,@00H ® Register 00H = 08H, register 01H = 0BFH
OR 00H,#02H ® Register 00H = 0AH

In the first example, if working register R0 contains the value 15H and register R1 the value 2AH, the statement "OR R0,R1" logical-ORs the R0 and R1 register contents and stores the result (3FH) in destination register R0.

The other examples show the use of the logical OR instruction with the various addressing modes and formats.

POP — Pop From Stack

POP dst

Operation: dst _ @SP

SP _ SP + 1

The contents of the location addressed by the stack pointer are loaded into the destination. The stack pointer is then incremented by one.

Flags: No flags affected.

Format:

opc	dst

Bytes	Cycles	Opcode (Hex)	Addr Mode dst
2	8	50	R
	8	51	IR

Examples: Given: Register 00H = 01H, register 01H = 1BH, SP (0D9H) = 0BBH, and stack register 0BBH = 55H:

POP 00H ® Register 00H = 55H, SP = 0BCH

POP @00H ® Register 00H = 01H, register 01H = 55H, SP = 0BCH

In the first example, general register 00H contains the value 01H. The statement "POP 00H" loads the contents of location 0BBH (55H) into destination register 00H and then increments the stack pointer by one. Register 00H then contains the value 55H and the SP points to location 0BCH.

PUSH — Push To Stack

PUSH src

Operation: SP _ SP – 1

@SP _ src

A PUSH instruction decrements the stack pointer value and loads the contents of the source (src) into the location addressed by the decremented stack pointer. The operation then adds the new value to the top of the stack.

Flags: No flags are affected.

Format:

	Bytes	Cycles	Opcode (Hex)	Addr Mode dst
opc | src	2	8	70	R
		8	71	IR

Examples: Given: Register 40H = 4FH, register 4FH = 0AAH, SP = 0C0H:

PUSH 40H ® Register 40H = 4FH, stack register 0BFH = 4FH, SP = 0BFH

PUSH @40H ® Register 40H = 4FH, register 4FH = 0AAH, stack register 0BFH = 0AAH, SP = 0BFH

In the first example, if the stack pointer contains the value 0C0H, and general register 40H the value 4FH, the statement "PUSH 40H" decrements the stack pointer from 0C0 to 0BFH. It then loads the contents of register 40H into location 0BFH. Register 0BFH then contains the value 4FH and SP points to location 0BFH.

RCF — Reset Carry Flag

RCF RCF

Operation: C _ 0

The carry flag is cleared to logic zero, regardless of its previous value.

Flags: **C:** Cleared to "0".

No other flags are affected.

Format:

	Bytes	Cycles	Opcode (Hex)
opc	1	4	CF

Example: Given: C = "1" or "0":

The instruction RCF clears the carry flag (C) to logic zero.

RET — Return

RET

Operation: PC _ @SP

SP _ SP + 2

The RET instruction is normally used to return to the previously executing procedure at the end of a procedure entered by a CALL instruction. The contents of the location addressed by the stack pointer are popped into the program counter. The next statement that is executed is the one that is addressed by the new program counter value.

Flags: No flags are affected.

Format:

	Bytes	Cycles	Opcode (Hex)
opc	1	8	AF
		10	

Example: Given: SP = 0BCH, (SP) = 101AH, and PC = 1234:

RET ® PC = 101AH, SP = 0BEH

The statement "RET" pops the contents of stack pointer location 0BCH (10H) into the high byte of the program counter. The stack pointer then pops the value in location 0BDH (1AH) into the PC's low byte and the instruction at location 101AH is executed. The stack pointer now points to memory location 0BEH.

RL — Rotate Left

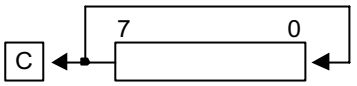
RL dst

Operation: C _ dst (7)

dst (0) _ dst (7)

dst (n + 1) _ dst (n), n = 0–6

The contents of the destination operand are rotated left one bit position. The initial value of bit 7 is moved to the bit zero (LSB) position and also replaces the carry flag.



Flags:
C: Set if the bit rotated from the most significant bit position (bit 7) was "1".
Z: Set if the result is "0"; cleared otherwise.
S: Set if the result bit 7 is set; cleared otherwise.
V: Set if arithmetic overflow occurred, that is, if the sign of the destination changed during rotation; cleared otherwise.

Format:

	Bytes	Cycles	Opcode (Hex)	Addr Mode dst
opc | dst	2	4	90	R
		4	91	IR

Examples: Given: Register 00H = 0AAH, register 01H = 02H and register 02H = 17H:

RL 00H ® Register 00H = 55H, C = "1"

RL @01H ® Register 01H = 02H, register 02H = 2EH, C = "0"

In the first example, if general register 00H contains the value 0AAH (10101010B), the statement "RL 00H" rotates the 0AAH value left one bit position, leaving the new value 55H (01010101B) and setting the carry and overflow flags.

RLC — Rotate Left Through Carry

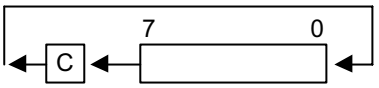
RLC dst

Operation: dst (0) _ C

C _ dst (7)

dst (n + 1) _ dst (n), n = 0–6

The contents of the destination operand with the carry flag are rotated left one bit position. The initial value of bit 7 replaces the carry flag (C); the initial value of the carry flag replaces bit zero.



Flags:
C: Set if the bit rotated from the most significant bit position (bit 7) was "1".
Z: Set if the result is "0"; cleared otherwise.
S: Set if the result bit 7 is set; cleared otherwise.
V: Set if arithmetic overflow occurred, that is, if the sign of the destination changed during rotation; cleared otherwise.

Format:

	Bytes	Cycles	Opcode (Hex)	Addr Mode dst
opc | dst	2	4	10	R
		4	11	IR

Examples: Given: Register 00H = 0AAH, register 01H = 02H, and register 02H = 17H, C = "0":

RLC 00H ® Register 00H = 54H, C = "1"

RLC @01H ® Register 01H = 02H, register 02H = 2EH, C = "0"

In the first example, if general register 00H has the value 0AAH (10101010B), the statement "RLC 00H" rotates 0AAH one bit position to the left. The initial value of bit 7 sets the carry flag and the initial value of the C flag replaces bit zero of register 00H, leaving the value 55H (01010101B). The MSB of register 00H resets the carry flag to "1" and sets the overflow flag.

RR — Rotate Right

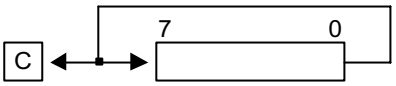
RR dst

Operation: C _ dst (0)

dst (7) _ dst (0)

dst (n) _ dst (n + 1), n = 0–6

The contents of the destination operand are rotated right one bit position. The initial value of bit zero (LSB) is moved to bit 7 (MSB) and also replaces the carry flag (C).



Flags:
C: Set if the bit rotated from the least significant bit position (bit zero) was "1".
Z: Set if the result is "0"; cleared otherwise.
S: Set if the result bit 7 is set; cleared otherwise.
V: Set if arithmetic overflow occurred, that is, if the sign of the destination changed during rotation; cleared otherwise.

Format:

	Bytes	Cycles	Opcode (Hex)	Addr Mode dst
opc | dst	2	4	E0	R
		4	E1	IR

Examples: Given: Register 00H = 31H, register 01H = 02H, and register 02H = 17H:

RR 00H ® Register 00H = 98H, C = "1"

RR @01H ® Register 01H = 02H, register 02H = 8BH, C = "1"

In the first example, if general register 00H contains the value 31H (00110001B), the statement "RR 00H" rotates this value one bit position to the right. The initial value of bit zero is moved to bit 7, leaving the new value 98H (10011000B) in the destination register. The initial bit zero also resets the C flag to "1" and the sign flag and overflow flag are also set to "1".

RRC — Rotate Right Through Carry

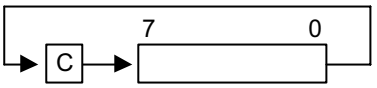
RRC dst

Operation: dst (7) _ C

C _ dst (0)

dst (n) _ dst (n + 1), n = 0–6

The contents of the destination operand and the carry flag are rotated right one bit position. The initial value of bit zero (LSB) replaces the carry flag; the initial value of the carry flag replaces bit 7 (MSB).



Flags:
C: Set if the bit rotated from the least significant bit position (bit zero) was "1".
Z: Set if the result is "0" cleared otherwise.
S: Set if the result bit 7 is set; cleared otherwise.
V: Set if arithmetic overflow occurred, that is, if the sign of the destination changed during rotation; cleared otherwise.

Format:

	Bytes	Cycles	Opcode (Hex)	Addr Mode dst
opc dst	2	4	C0	R
		4	C1	IR

Examples: Given: Register 00H = 55H, register 01H = 02H, register 02H = 17H, and C = "0":

RRC 00H ® Register 00H = 2AH, C = "1"

RRC @01H ® Register 01H = 02H, register 02H = 0BH, C = "1"

In the first example, if general register 00H contains the value 55H (01010101B), the statement "RRC 00H" rotates this value one bit position to the right. The initial value of bit zero ("1") replaces the carry flag and the initial value of the C flag ("1") replaces bit 7. This leaves the new value 2AH (00101010B) in destination register 00H. The sign flag and overflow flag are both cleared to "0".

SBC — Subtract With Carry

SBC dst,src

Operation: dst _ dst – src – c

The source operand, along with the current value of the carry flag, is subtracted from the destination operand and the result is stored in the destination. The contents of the source are unaffected. Subtraction is performed by adding the two's-complement of the source operand to the destination operand. In multiple precision arithmetic, this instruction permits the carry ("borrow") from the subtraction of the low-order operands to be subtracted from the subtraction of high-order operands.

Flags:

C: Set if a borrow occurred (src > dst); cleared otherwise.
Z: Set if the result is "0"; cleared otherwise.
S: Set if the result is negative; cleared otherwise.
V: Set if arithmetic overflow occurred, that is, if the operands were of opposite sign and the sign of the result is the same as the sign of the source; cleared otherwise.

Format:

	Bytes	Cycles	Opcode (Hex)	Addr Mode dst	Addr Mode src
opc | dst | src	2	4	32	r	r
		6	33	r	lr
opc | src | dst	3	6	34	R	R
		6	35	R	IR
opc | dst | src	3	6	36	R	IM

Examples: Given: R1 = 10H, R2 = 03H, C = "1", register 01H = 20H, register 02H = 03H, and register 03H = 0AH:

SBC	R1,R2	®	R1 = 0CH, R2 = 03H
SBC	R1,@R2	®	R1 = 05H, R2 = 03H, register 03H = 0AH
SBC	01H,02H	®	Register 01H = 1CH, register 02H = 03H
SBC	01H,@02H	®	Register 01H = 15H,register 02H = 03H, register 03H = 0AH
SBC	01H,#8AH	®	Register 01H = 95H; C, S, and V = "1"

In the first example, if working register R1 contains the value 10H and register R2 the value 03H, the statement "SBC R1,R2" subtracts the source value (03H) and the C flag value ("1") from the destination (10H) and then stores the result (0CH) in register R1.

SCF — Set Carry Flag

SCF

Operation: C _ 1

The carry flag (C) is set to logic one, regardless of its previous value.

Flags: **C:** Set to "1".

No other flags are affected.

Format:

	Bytes	Cycles	Opcode (Hex)
opc	1	4	DF

Example: The statement

SCF

sets the carry flag to logic one.

SRA — Shift Right Arithmetic

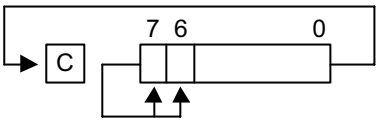
SRA dst

Operation: dst (7) _ dst (7)

C _ dst (0)

dst (n) _ dst (n + 1), n = 0–6

An arithmetic shift-right of one bit position is performed on the destination operand. Bit zero (the LSB) replaces the carry flag. The value of bit 7 (the sign bit) is unchanged and is shifted into bit position 6.



Flags:
C: Set if the bit shifted from the LSB position (bit zero) was "1".
Z: Set if the result is "0"; cleared otherwise.
S: Set if the result is negative; cleared otherwise.
V: Always cleared to "0".

Format:

	Bytes	Cycles	Opcode (Hex)	Addr Mode dst
opc dst	2	4	D0	R
		4	D1	IR

Examples: Given: Register 00H = 9AH, register 02H = 03H, register 03H = 0BCH, and C = "1":

SRA 00H ® Register 00H = 0CD, C = "0"

SRA @02H ® Register 02H = 03H, register 03H = 0DEH, C = "0"

In the first example, if general register 00H contains the value 9AH (10011010B), the statement "SRA 00H" shifts the bit values in register 00H right one bit position. Bit zero ("0") clears the C flag and bit 7 ("1") is then shifted into the bit 6 position (bit 7 remains unchanged). This leaves the value 0CDH (11001101B) in destination register 00H.

STOP — Stop Operation

STOP

Operation: The STOP instruction stops the both the CPU clock and system clock and causes the microcontroller to enter Stop mode. During Stop mode, the contents of on-chip CPU registers, peripheral registers, and I/O port control and data registers are retained. Stop mode can be released by an external reset operation or External interrupt input. For the reset operation, the RESET pin must be held to Low level until the required oscillation stabilization interval has elapsed.

Flags: No flags are affected.

Format:

	Bytes	Cycles	Opcode (Hex)	Addr Mode dst	src
opc	1	4	7F	–	–

Example: The statement

```
LD      STOPCON, #0A5H
STOP
NOP
NOP
NOP
```

halts all microcontroller operations. When STOPCON register is not #0A5H value, if you use STOP instruction, PC is changed to reset address.

SUB — Subtract

SUB dst,src

Operation: dst _ dst – src

The source operand is subtracted from the destination operand and the result is stored in the destination. The contents of the source are unaffected. Subtraction is performed by adding the two's complement of the source operand to the destination operand.

Flags:

C: Set if a "borrow" occurred; cleared otherwise.
Z: Set if the result is "0"; cleared otherwise.
S: Set if the result is negative; cleared otherwise.
V: Set if arithmetic overflow occurred, that is, if the operands were of opposite signs and the sign of the result is of the same as the sign of the source operand; cleared otherwise.

Format:

	Bytes	Cycles	Opcode (Hex)	Addr Mode dst	src
opc | dst | src	2	4	22	r	r
		6	23	r	lr
opc | src | dst	3	6	24	R	R
		6	25	R	IR
opc | dst | src	3	6	26	R	IM

Examples: Given: R1 = 12H, R2 = 03H, register 01H = 21H, register 02H = 03H, register 03H = 0AH:

SUB R1,R2 ® R1 = 0FH, R2 = 03H
SUB R1,@R2 ® R1 = 08H, R2 = 03H
SUB 01H,02H ® Register 01H = 1EH, register 02H = 03H
SUB 01H,@02H ® Register 01H = 17H, register 02H = 03H
SUB 01H,#90H ® Register 01H = 91H; C, S, and V = "1"
SUB 01H,#65H ® Register 01H = 0BCH; C and S = "1", V = "0"

In the first example, if working register R1 contains the value 12H and if register R2 contains the value 03H, the statement "SUB R1,R2" subtracts the source value (03H) from the destination value (12H) and stores the result (0FH) in destination register R1.

TCM — Test Complement Under Mask

TCM dst,src

Operation: (NOT dst) AND src

This instruction tests selected bits in the destination operand for a logic one value. The bits to be tested are specified by setting a "1" bit in the corresponding position of the source operand (mask). The TCM statement complements the destination operand, which is then ANDed with the source mask. The zero (Z) flag can then be checked to determine the result. The destination and source operands are unaffected.

Flags:
C: Unaffected.
Z: Set if the result is "0"; cleared otherwise.
S: Set if the result bit 7 is set; cleared otherwise.
V: Always cleared to "0".

Format:

	Bytes	Cycles	Opcode (Hex)	Addr Mode dst	Addr Mode src
opc | dst | src	2	4	62	r	r
		6	63	r	lr
opc | src | dst	3	6	64	R	R
		6	65	R	IR
opc | dst | src	3	6	66	R	IM

Examples: Given: R0 = 0C7H, R1 = 02H, R2 = 12H, register 00H = 2BH, register 01H = 02H, and register 02H = 23H:

TCM	R0,R1	®	R0 = 0C7H, R1 = 02H, Z = "1"
TCM	R0,@R1	®	R0 = 0C7H, R1 = 02H, register 02H = 23H, Z = "0"
TCM	00H,01H	®	Register 00H = 2BH, register 01H = 02H, Z = "1"
TCM	00H,@01H	®	Register 00H = 2BH, register 01H = 02H, register 02H = 23H, Z = "1"
TCM	00H,#34	®	Register 00H = 2BH, Z = "0"

In the first example, if working register R0 contains the value 0C7H (11000111B) and register R1 the value 02H (00000010B), the statement "TCM R0,R1" tests bit one in the destination register for a "1" value. Because the mask value corresponds to the test bit, the Z flag is set to logic one and can be tested to determine the result of the TCM operation.

TM — Test Under Mask

TM dst,src

Operation: dst AND src

This instruction tests selected bits in the destination operand for a logic zero value. The bits to be tested are specified by setting a "1" bit in the corresponding position of the source operand (mask), which is ANDed with the destination operand. The zero (Z) flag can then be checked to determine the result. The destination and source operands are unaffected.

Flags:
C: Unaffected.
Z: Set if the result is "0"; cleared otherwise.
S: Set if the result bit 7 is set; cleared otherwise.
V: Always reset to "0".

Format:

	Bytes	Cycles	Opcode (Hex)	Addr Mode dst	Addr Mode src
opc | dst | src	2	4	72	r	r
		6	73	r	lr
opc | src | dst	3	6	74	R	R
		6	75	R	IR
opc | dst | src	3	6	76	R	IM

Examples: Given: R0 = 0C7H, R1 = 02H, R2 = 18H, register 00H = 2BH, register 01H = 02H, and register 02H = 23H:

TM	R0,R1	®	R0 = 0C7H, R1 = 02H, Z = "0"
TM	R0,@R1	®	R0 = 0C7H, R1 = 02H, register 02H = 23H, Z = "0"
TM	00H,01H	®	Register 00H = 2BH, register 01H = 02H, Z = "0"
TM	00H,@01H	®	Register 00H = 2BH, register 01H = 02H, register 02H = 23H, Z = "0"
TM	00H,#54H	®	Register 00H = 2BH, Z = "1"

In the first example, if working register R0 contains the value 0C7H (11000111B) and register R1 the value 02H (00000010B), the statement "TM R0,R1" tests bit one in the destination register for a "0" value. Because the mask value does not match the test bit, the Z flag is cleared to logic zero and can be tested to determine the result of the TM operation.

XOR — Logical Exclusive OR

XOR dst,src

Operation: dst _ dst XOR src

The source operand is logically exclusive-ORed with the destination operand and the result is stored in the destination. The exclusive-OR operation results in a "1" bit being stored whenever the corresponding bits in the operands are different; otherwise, a "0" bit is stored.

Flags:
C: Unaffected.
Z: Set if the result is "0"; cleared otherwise.
S: Set if the result bit 7 is set; cleared otherwise.
V: Always reset to "0".

Format:

	Bytes	Cycles	Opcode (Hex)	Addr Mode dst	src
opc | dst | src	2	4	B2	r	r
		6	B3	r	lr
opc | src | dst	3	6	B4	R	R
		6	B5	R	IR
opc | dst | src	3	6	B6	R	IM

Examples: Given: R0 = 0C7H, R1 = 02H, R2 = 18H, register 00H = 2BH, register 01H = 02H, and register 02H = 23H:

XOR R0,R1 ® R0 = 0C5H, R1 = 02H
XOR R0,@R1 ® R0 = 0E4H, R1 = 02H, register 02H = 23H
XOR 00H,01H ® Register 00H = 29H, register 01H = 02H
XOR 00H,@01H ® Register 00H = 08H, register 01H = 02H, register 02H = 23H
XOR 00H,#54H ® Register 00H = 7FH

In the first example, if working register R0 contains the value 0C7H and if register R1 contains the value 02H, the statement "XOR R0,R1" logically exclusive-ORs the R1 value with the R0 value and stores the result (0C5H) in the destination register R0.

7 CLOCK CIRCUIT

OVERVIEW

The clock frequency generation for the S3C9484/C9488/F9488 by an external crystal can range from 1 MHz to 8 MHz. The maximum CPU clock frequency is 8 MHz. The X_{IN} and X_{OUT} pins connect the external oscillator or clock source to the on-chip clock circuit.

SYSTEM CLOCK CIRCUIT

The system clock circuit has the following components:

— External crystal or ceramic resonator oscillation source (or an external clock source)
— Oscillator stop and wake-up functions
— Programmable frequency divider for the CPU clock (fxx divided by 1, 2, 8, or 16)
— System clock control register, CLKCON
— Oscillator control register, OSCCON and STOP control register, STPCON

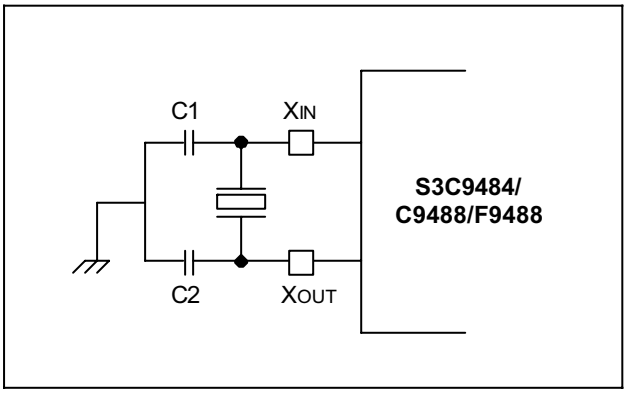


Figure 7-1. Main Oscillator Circuit (Crystal or Ceramic Oscillator)

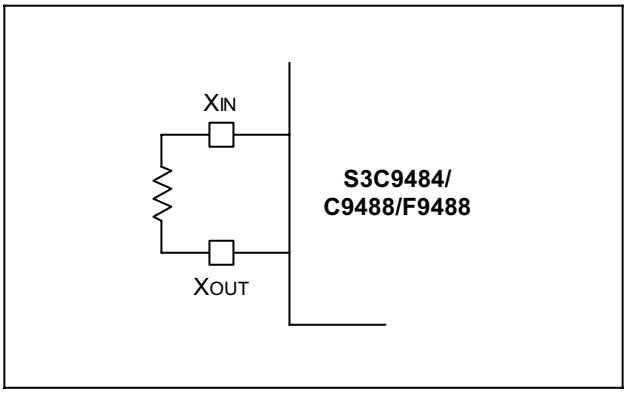


Figure 7-2. Main Oscillator Circuit (RC Oscillator)

CLOCK STATUS DURING POWER-DOWN MODES

The two power-down modes, Stop mode and Idle mode, affect the system clock as follows:

— In Stop mode, the main oscillator is halted. Stop mode is released, and the oscillator started, by a reset operation or an external interrupt (with RC delay noise filter), and can be released by internal interrupt too when the sub-system oscillator is running and watch timer is operating with sub-system clock.

— In Idle mode, the internal clock signal is gated to the CPU, but not to interrupt structure, timers and timer/counters. Idle mode is released by a reset or by an external or internal interrupt.

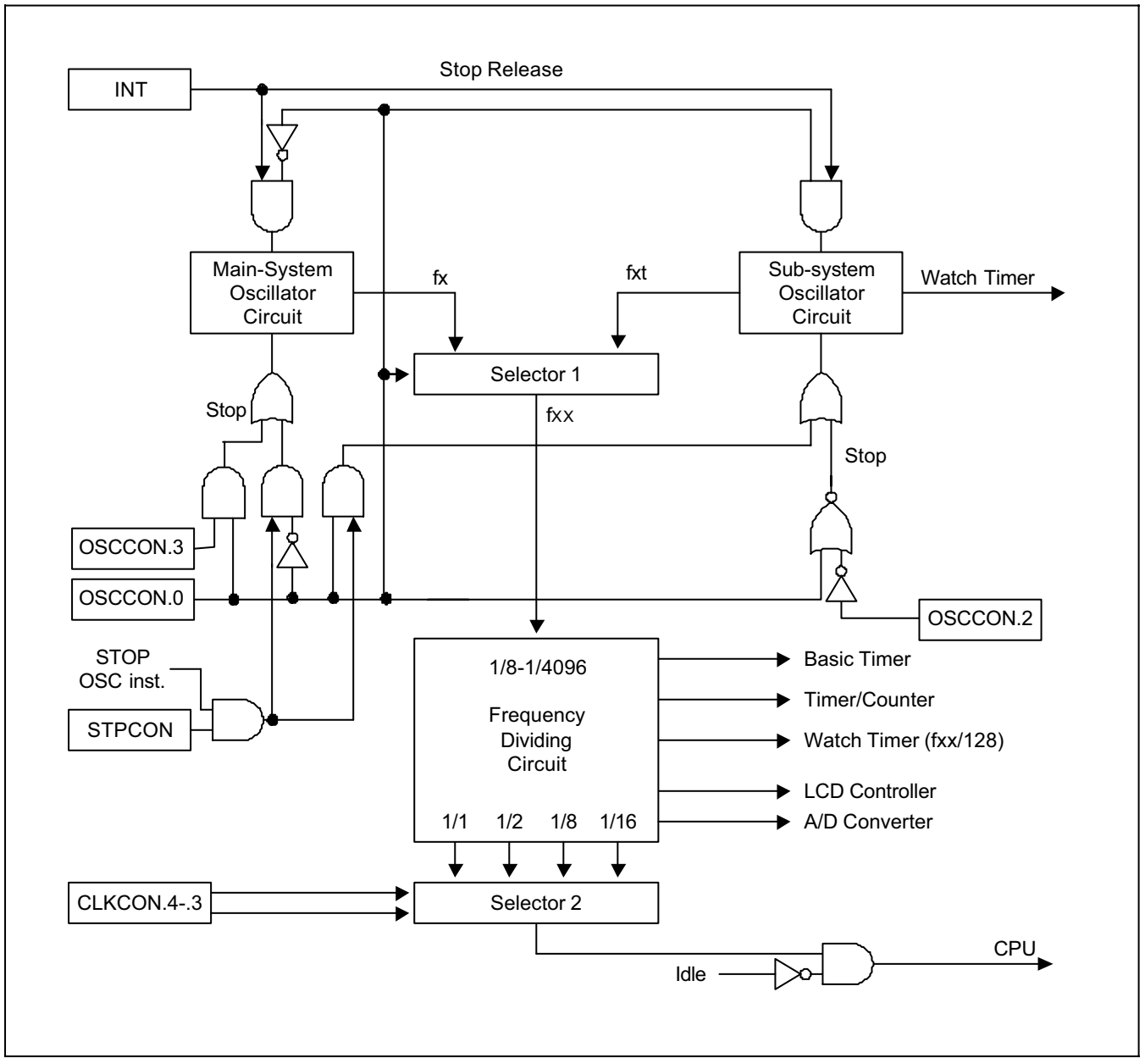


Figure 7-3. System Clock Circuit Diagram

SYSTEM CLOCK CONTROL REGISTER (CLKCON)

The system clock control register, CLKCON, is located at address D4H. It is read/write addressable and has the following functions:

— Oscillator frequency divide-by value

After the main oscillator is activated, and the fxx/16 (the slowest clock speed) is selected as the CPU clock. If necessary, you can then increase the CPU clock speed to fxx/8, fxx/2, or fxx/1.

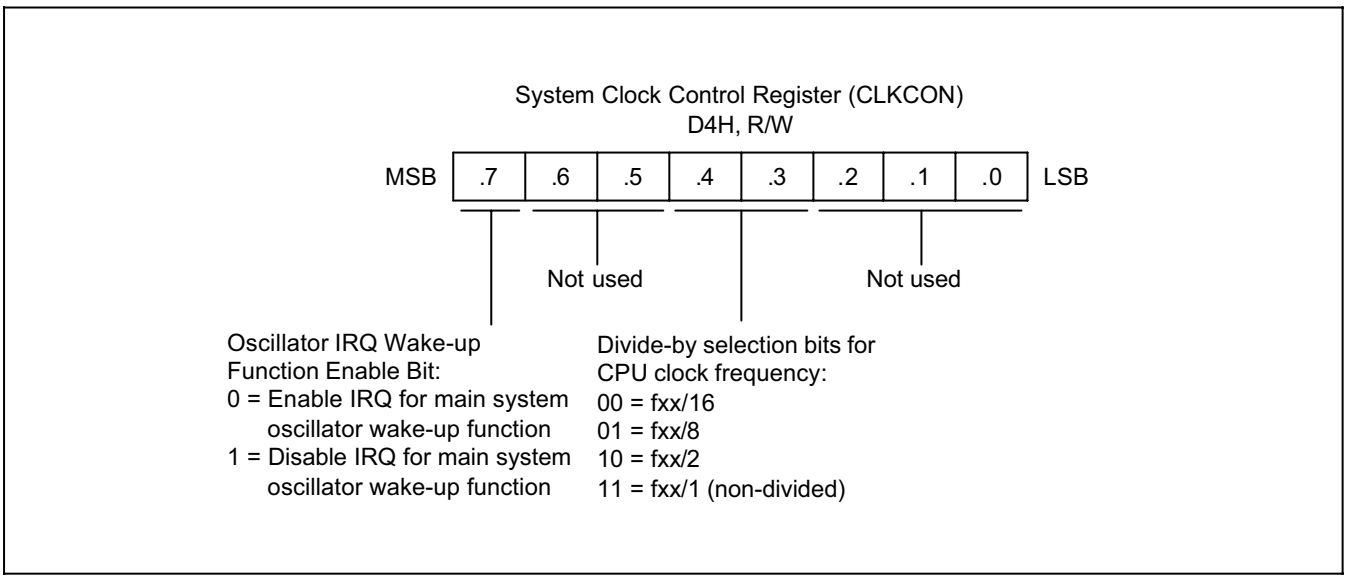
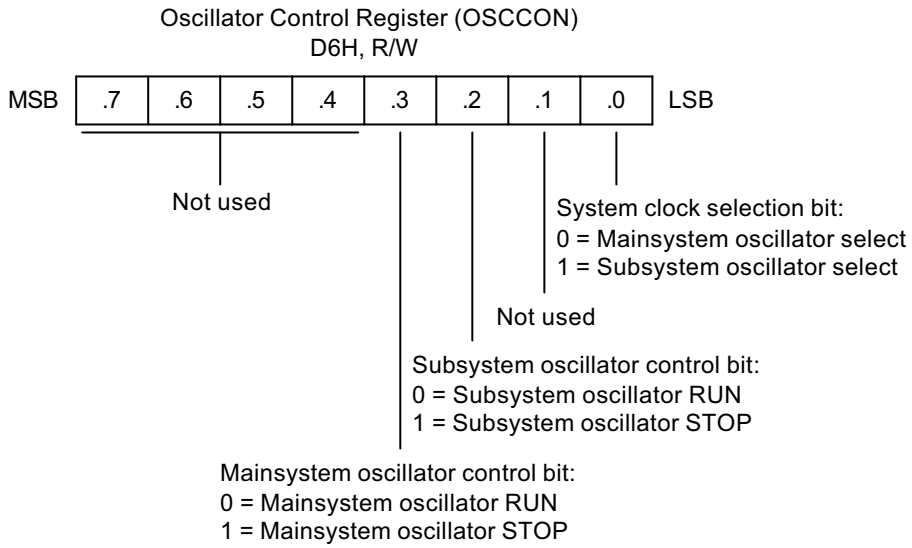


Figure 7-4. System Clock Control Register (CLKCON)

MAIN/SUBSYSTEM OSCILLATOR SELECTION (OSCCON)

When a main oscillator is selected, users cannot stop operating of a main oscillator by handling the OSCCON register but sub oscillator can be stopped. If users intend to stop operating of a main oscillator users must use "STOP" instruction.
When a sub oscillator is selected, users must do the contrary of the above case.

NOTE: If a sub oscillator is not used, users must connect it to Vss.



NOTE: When the CPU is operated with fxt (sub-oscillation clock), it is possible to use the stop instruction but in this case before using stop instruction***you must select fxx/128 for basic timer counter input clock***. Then the oscillation stabilization time is 62.5 ((1/32768) x 128 x 16) ms. Here the warm-up time is from the stop release signal activates until the basic timer counter counting start. So the totaly needed oscillation stabilization time will be less than 162.5 ms.

Figure 7-5. Oscillator Control Register (OSCCON)

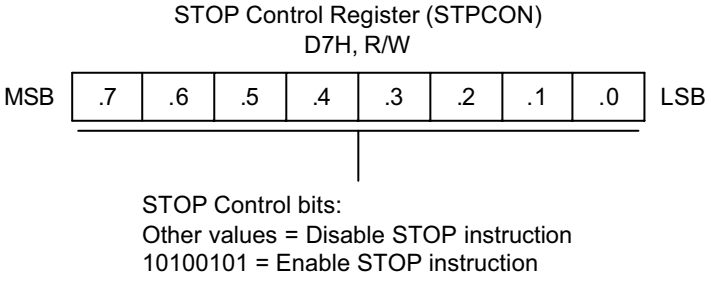


Figure 7-6. STOP Control Register (STPCON)

8 RESET and POWER-DOWN

SYSTEM RESET

OVERVIEW

During a power-on reset, the voltage at V_{DD} goes to High level and the RESET pin is forced to Low level. The RESET signal is input through a Schmitt trigger circuit where it is then synchronized with the CPU clock. This procedure brings S3C9484/C9488/F9488 into a known operating status.

To allow time for internal CPU clock oscillation to stabilize, the RESET pin must be held to Low level for a minimum time interval after the power supply comes within tolerance. The minimum required oscillation stabilization time for a reset operation is 1millisecond.

Whenever a reset occurs during normal operation (that is, when both V_{DD} and RESET are High level), the RESET pin is forced Low and the reset operation starts. All system and peripheral control registers are then reset to their default hardware values.

In summary, the following sequence of events occurs during a reset operation:

— Interrupt is disabled.

— The watchdog function is enabled.

— Ports 0-4 are set to input mode. (except P0.0-2, P3.3-6)

— Peripheral control and data registers are disabled and reset to their default hardware values.

— The program counter (PC) is loaded with the program reset address in the ROM, 0100H.

— When the programmed oscillation stabilization time interval has elapsed, the instruction stored in ROM location 0100H (and 0101H) is fetched and executed.

NORMAL MODE RESET OPERATION

In normal (masked ROM) mode, the Test pin is tied to V_{SS}. A reset enables access to the 4/8-Kbyte on-chip ROM. (The external interface is not automatically configured).

NOTE

To program the duration of the oscillation stabilization interval, you make the appropriate settings to the basic timer control register, BTCON, *before* entering Stop mode. Also, if you do not want to use the watchdog timer function (which causes a system reset if a watchdog timer counter overflow occurs), you can disable it by writing '1010B' to the upper nibble of WDTCON.

HARDWARE RESET VALUES

The reset values for CPU and system registers, peripheral control registers, and peripheral data registers following a reset operation. The following notation is used to represent reset values:

— A "1" or a "0" shows the reset bit value as logic one or logic zero, respectively.

— An "x" means that the bit value is undefined after a reset.

— A dash ("–") means that the bit is either not used or not mapped, but read 0 is the bit value.

Table 8-1. S3C9484/C9488/F9488 Register Values after RESET

Register Name	Mnemonic	Address		Bit Values After RESET							
		Dec	Hex	7	6	5	4	3	2	1	0
LCD control register	LCDCON	208	D0H	0	–	0	0	0	0	0	0
LCD drive voltage control register	LCDVOL	209	D1H	0	–	–	–	0	0	0	0
Port 0 pull-up resistor control register	P0PUR	210	D2H	1	1	1	1	1	1	1	1
Port 1 pull-up resistor control register	P1PUR	211	D3H	1	1	1	1	1	1	1	1
System Clock control register	CLKCON	212	D4H	0	–	–	0	0	–	–	–
System flags register	FLAGS	213	D5H	x	x	x	x	–	–	–	–
Oscillator control register	OSCCON	214	D6H	–	–	–	–	0	0	–	0
STOP control register	STPCON	215	D7H	0	0	0	0	0	0	0	0
Voltage Level Detector control register	VLDCON	216	D8H	–	0	1	0	1	1	0	0
Stack pointer register	SP	217	D9H	x	x	x	x	x	x	x	x
Location DAH-DBH are not mapped											
Basic timer control register	BTCON	220	DCH	–	–	–	–	0	0	0	0
Basic timer counter register	BTCNT	221	DDH	0	0	0	0	0	0	0	0
Location DEH is not mapped											
System mode register	SYM	223	DFH	–	–	–	–	0	0	0	0
Port 0 Data Register	P0	224	E0H	0	0	0	0	0	0	0	0
Port 1 Data Register	P1	225	E1H	0	0	0	0	0	0	0	0
Port 2 Data Register	P2	226	E2H	0	0	0	0	0	0	0	0
Port 3 Data Register	P3	227	E3H	0	0	0	0	0	0	0	0
Port 4 Data Register	P4	228	E4H	0	0	0	0	0	0	0	0
Watchdog timer control register	WDTCON	229	E5H	0	0	0	0	0	0	0	0
Port 0 control High register	P0CONH	230	E6H	0	0	0	0	0	0	0	0
Port 0 control Low register	P0CONL	231	E7H	0	0	0	0	0	0	0	0
Port 1 control High register	P1CONH	232	E8H	0	0	0	0	0	0	0	0
Port 1 control Low register	P1CONL	233	E9H	0	0	0	0	0	0	0	0

Table 8-1. S3C9484/C9488/F9488 Registers Values after RESET (continued)

Register Name	Mnemonic	Address		Bit Values After RESET							
		Dec	Hex	7	6	5	4	3	2	1	0
Port 2 control High register	P2CONH	234	EAH	0	0	0	0	0	0	0	0
Port 2 control Low register	P2CONL	235	EBH	0	0	0	0	0	0	0	0
Port 3 control High register	P3CONH	236	ECH	S	S	S	S	S	S	S	S
Port 3 control Low register	P3CONL	237	EDH	0	0	0	0	0	0	0	0
Port 3 interrupt control register	P3INT	238	EEH	0	0	0	0	0	0	0	0
Port 3 interrupt pending register	P3PND	239	EFH	–	–	–	–	0	0	0	0
Port 4 control High register	P4CONH	240	F0H	–	–	0	0	0	0	0	0
Port 4 control Low register	P4CONL	241	F1H	0	0	0	0	0	0	0	0
Timer A/B interrupt pending register	TINTPND	242	F2H	–	–	–	–	–	0	0	0
Timer A control register	TACON	243	F3H	0	0	0	0	0	0	0	0
Timer A counter register	TACNT	244	F4H	0	0	0	0	0	0	0	0
Timer A data register	TADATA	245	F5H	1	1	1	1	1	1	1	1
Timer B data register(high byte)	TBDATAH	246	F6H	1	1	1	1	1	1	1	1
Timer B data register(low byte)	TBDATAL	247	F7H	1	1	1	1	1	1	1	1
Timer B control register	TBCON	248	F8H	0	0	0	0	0	0	0	0
Watch timer control register	WTCON	249	F9H	0	0	0	0	0	0	0	0
A/D converter data register(high byte)	ADDATAH	250	FAH	–	–	–	–	–	–	0	0
A/D converter data register(low byte)	ADDATAL	251	FBH	0	0	0	0	0	0	0	0
A/D converter control register	ADCON	252	FCH	0	0	0	0	0	0	0	0
UART control register	UARTCON	253	FDH	0	0	0	0	0	0	0	0
UART pending register	UARTPND	254	FEH	–	–	0	0	–	–	0	0
UART data register	UDATA	255	FFH	x	x	x	x	x	x	x	x

Table 8-2. S3C9484/C9488/F9488 Registers Values after RESET (page 1)

Register Name	Mnemonic	Address		Bit Values After RESET							
		Dec	Hex	7	6	5	4	3	2	1	0
UART baud rate data register(high byte)	BRDATAH	20	14H	1	1	1	1	1	1	1	1
UART baud rate data register(low byte)	BRDATAL	21	15H	1	1	1	1	1	1	1	1

NOTE: –: Not mapped or not used, x: Undefined, S: be set by Smart option.

POWER-DOWN MODES

STOP MODE

Stop mode is invoked by the instruction STOP (opcode 7FH). In Stop mode, the operation of the CPU and all peripherals is halted. That is, the on-chip main oscillator stops and the supply current is reduced to less than 3 μA. All system functions stop when the clock "freezes," but data stored in the internal register file is retained. Stop mode can be released in one of two ways: by a reset or by interrupts.

NOTE

Do not use stop mode if you are using an external clock source because X_{IN} input must be restricted internally to V_{SS} to reduce current leakage.

Using RESET to Release Stop Mode

Stop mode is released when the RESET signal is released and returns to high level: all system and peripheral control registers are reset to their default hardware values and the contents of all data registers are retained. A reset operation automatically selects a slow clock (1/16) because CLKCON.3 and CLKCON.4 are cleared to '00B'. After the programmed oscillation stabilization interval has elapsed, the CPU starts the system initialization routine by fetching the program instruction stored in ROM location 0100H (and 0101H).

Using an External Interrupt to Release Stop Mode

External interrupts with an RC-delay noise filter circuit can be used to release Stop mode. Which interrupt you can use to release Stop mode in a given situation depends on the microcontroller's current internal operating mode. The external interrupts in the S3C9484/C9488/F9488 interrupt structure that can be used to release Stop mode are:

— External interrupts P3.3-P3.6 (INT0-INT3)

Please note the following conditions for Stop mode release:

— If you release Stop mode using an external interrupt, the current values in system and peripheral control registers are unchanged except **STPCON register**.

— If you use an external interrupt for Stop mode release, you can also program the duration of the oscillation stabilization interval. To do this, you must make the appropriate control and clock settings *before* entering Stop mode.

— When the Stop mode is released by external interrupt, the CLKCON.4 and CLKCON.3 bit-pair setting remains unchanged and the currently selected clock value is used.

— The external interrupt is serviced when the Stop mode release occurs. Following the IRET from the service routine, the instruction immediately following the one that initiated Stop mode is executed.

Using an internal Interrupt to Release Stop Mode

If you use Watch Timer with sub oscillator, STOP mode is released by WATCH TIMER interrupt.

How to enter into stop mode

Handling STPCON register then writing STOP instruction. (Keep the order)

Attentions of Using Stop Mode

If you use 42-pin Package, you must set P0.3- P0.4 for output mode and must set out value on low.

And If you use 32-pin Package, you must set P4.0- P4.6/P0.3- P0.7 for output mode and must set out value to low to prevent the leaky current in stop mode.

IDLE MODE

Idle mode is invoked by the instruction IDLE (opcode 6FH). In idle mode, CPU operations are halted while some peripherals remain active. During idle mode, the internal clock signal is gated away from the CPU, but all peripherals timers remain active. Port pins retain the mode (input or output) they had at the time idle mode was entered.

There are two ways to release idle mode:

1. Execute a reset. All system and peripheral control registers are reset to their default values and the contents of all data registers are retained. The reset automatically selects the slow clock fxx/16 because CLKCON.4 and CLKCON.3 are cleared to '00B'. If interrupts are masked, a reset is the only way to release idle mode.
2. Activate any enabled interrupt, causing idle mode to be released. When you use an interrupt to release idle mode, the CLKCON.4 and CLKCON.3 register values remain unchanged, and the currently selected clock value is used. The interrupt is then serviced. When the return-from-interrupt (IRET) occurs, the instruction immediately following the one that initiated idle mode is executed.

NOTES

9 I/O PORTS

OVERVIEW

The S3C9484/C9488/F9488 microcontroller has five bit-programmable I/O ports, P0-P4. The port 3 and 4 are 7-bit ports and the others are 8-bit ports. This gives a total of 38 I/O pins. Each port can be flexibly configured to meet application design requirements. The CPU accesses ports by directly writing or reading port registers. No special I/O instructions are required.

Table 9-1 gives you a general overview of the S3C9484/C9488/F9488 I/O port functions.

Table 9-1. S3C9484/C9488/F9488 Port Configuration Overview

Port	Configuration Options
0	I/O port with bit-programmable pins. Configurable to input or push-pull output mode. Pull-up resistors can be assigned by software. Pins can also be assigned individually as alternative function pins.
1	I/O port with bit-programmable pins. Configurable to input or push-pull output mode. Pull-up resistors can be assigned by software. Pins can also be assigned individually as alternative function pins.
2	I/O port with bit-programmable pins. Configurable to input mode, push-pull output mode. Pins can also be assigned individually as alternative function pins.
3	I/O port with bit-programmable pins. Configurable to input mode, push-pull output mode. Pins can also be assigned individually as alternative function pins.
4	I/O port with bit-programmable pins. Configurable to input mode, push-pull output mode. Pins can also be assigned individually as alternative function pins.

PORT DATA REGISTERS

Table 9-2 gives you an overview of the register locations of all five S3C9484/C9488/F9488 I/O port data registers. Data registers for ports 0, 1, 2, 3, and 4 have the general format shown in Figure 9-1.

Table 9-2. Port Data Register Summary

Register Name	Mnemonic	Decimal	Hex	R/W
Port 0 data register	P0	224	E0H	R/W
Port 1 data register	P1	225	E1H	R/W
Port 2 data register	P2	226	E2H	R/W
Port 3 data register	P3	227	E3H	R/W
Port 4 data register	P4	228	E4H	R/W

PORT 0

Port 0 is an 8-bit I/O Port that you can use two ways:

— General-purpose I/O

— Alternative function

Port 0 is accessed directly by writing or reading the port 0 data register, P0 at location E0H.

Port 0 Control Register (P0CONH, P0CONL, P0PUR)

Port 0 pins are configured individually by bit-pair settings in three control registers located :
P0CONL (low byte, E7H) , P0CONH (high byte, E6H) and P0PUR (D2H).

When you select output mode, a push-pull circuit is configured. In input mode, many different selections are available:

— Input mode.

— Push-pull output mode

— Alternative function: LCD ‘COM’ signal output – COM4, COM5, COM6, COM7

— Alternative function: ADC input mode – ADC4, ADC5, ADC6, ADC7, ADC8

— Alternative function: RESETB

— Alternative function: Xtin/Xtout

Port 0 Control Register, High Byte (P0CONH)
E6H, R/W, Reset value:00H

MSB	.7	.6	.5	.4	.3	.2	.1	.0	LSB
	P0.7 COM4/ ADC4		P0.6 COM5/ ADC5		P0.5 COM6/ ADC6		P0.4 /COM7 /ADC7		

.7 .6	
0 0	Input mode
0 1	Alternative function: ADC4 Input
1 0	Push-pull output
1 1	Alternative function: LCD COM4 signal output

.5 .4	
0 0	Input mode
0 1	Alternative function: ADC5 Input
1 0	Push-pull output
1 1	Alternative function: LCD COM5 signal output

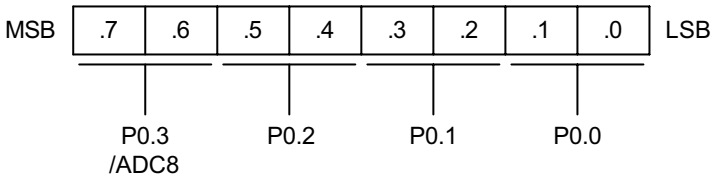
.3 .2	
0 0	Input mode
0 1	Alternative function: ADC6 Input
1 0	Push-pull output
1 1	Alternative function: LCD COM6 signal output

.1 .0	
0 0	Input mode
0 1	Alternative function: ADC7 Input
1 0	Push-pull output
1 1	Alternative function: LCD COM7 signal output

NOTE: You must be care of the pull-up resistor option.

Figure 9-1. Port 0 High-Byte Control Register (P0CONH)

Port 0 Control Register, Low Byte (P0CONL)
E7H, R/W, Reset value:00H



.7 .6	
0 x	Input mode
1 0	Push-pull output
1 1	Alternative function: ADC8 input

.5 .4	
0 x	Input mode
1 x	Push-pull output

.3 .2	
0 x	Input mode
1 x	Push-pull output

.1 .0	
0 x	Input mode
1 x	Push-pull output

NOTES:

1. You must determine P0.0-P0.2 function on smart option.
   In other word, After reset operation, you cann't change P0.0-.2 function.
   If you selected Normal I/O function at smart option,
   After reset operation, you can use on Normal I/O and you can control P0.0-.2 by this control register value.
2. You must be care of the pull-up resistor option.

Figure 9-2. Port 0 Low-Byte Control Register (P0CONL)

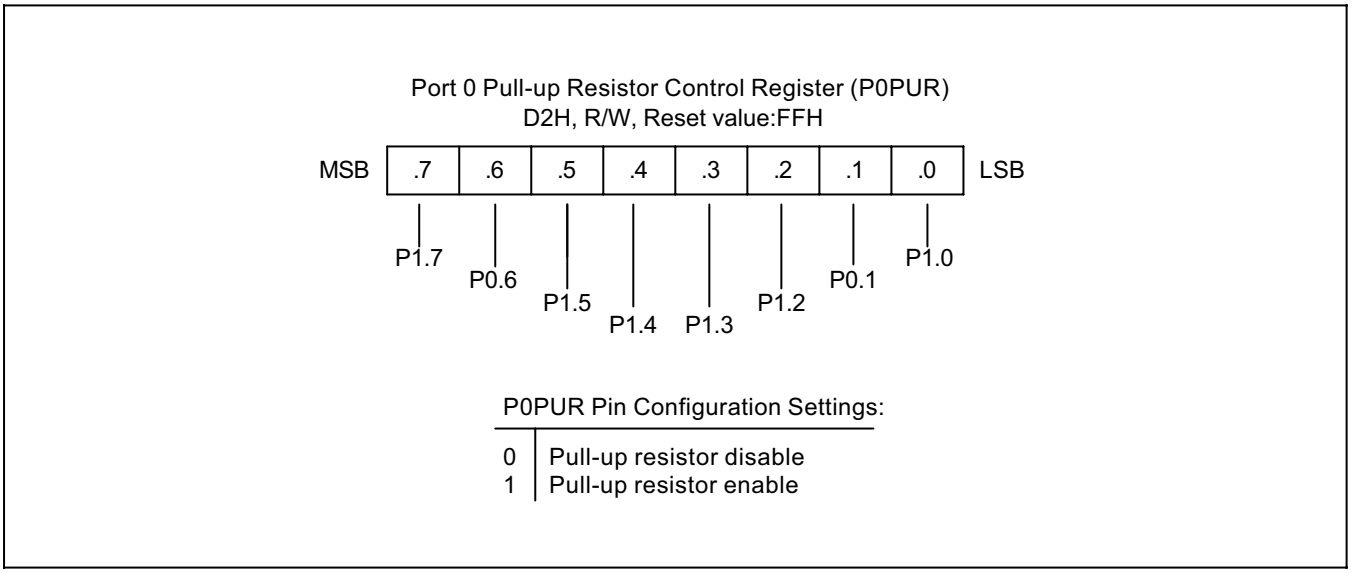


Figure 9-3. Port 0 Pull-up Resistor Control Register (P0PUR)

PORT 1

Port 1 is an 8-bit I/O port that you can use two ways:

— General-purpose I/O

— Alternative function

Port 1 is accessed directly by writing or reading the port 1 data register, P1 at location E1H.

Port 1 Control Register (P1CONH, P1CONL, P1PUR)

Port 1 pins are configured individually by bit-pair settings in three control registers located: P1CONL(low byte, E9H), P1CONH(high byte, E8H) and P1PUR(D3H).

When you select output mode, a push-pull circuit is configured. In input mode, many different selections are available:

— Input mode.

— Push-pull output mode

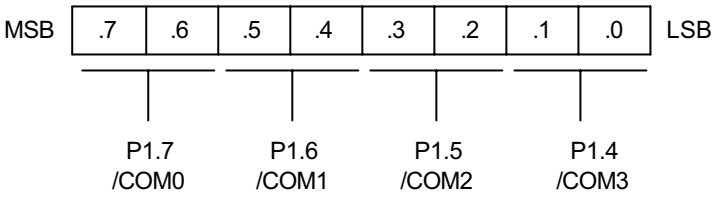
— Alternative function: LCD ‘COM’ signal output – COM0, COM1, COM2, COM3

— Alternative function: TBPWM output

— Alternative function: BUZ output

— Alternative function: ADC input mode – ADC0, ADC1, ADC2, ADC3

Port 1 Control Register, High Byte (P1CONH)
E8H, R/W, Reset value:00H



P1.7 /COM0	P1.6 /COM1	P1.5 /COM2	P1.4 /COM3

.7 .6	
0 x	Input mode
1 0	Push-pull output
1 1	Alternative function: LCD COM0 signal output

.5 .4	
0 x	Input mode
1 0	Push-pull output
1 1	Alternative function: LCD COM1 signal output

.3 .2	
0 x	Input mode
1 0	Push-pull output
1 1	Alternative function: LCD COM2 signal output

.1 .0	
0 x	Input mode
1 0	Push-pull output
1 1	Alternative function: LCD COM3signal output

NOTE: You must be care of the pull-up resistor option.

Figure 9-4. Port 1 High-Byte Control Register (P1CONH)

Port 1 Control Register, Low Byte (P1CONL)
E9H, R/W, Reset value:00H

MSB	.7	.6	.5	.4	.3	.2	.1	.0	LSB

P1.3 /ADC0	P1.2 /ADC1	P1.1 /ADC2 /BUZ	P1.0 /ADC3 /TBPWM

.7 .6	
0 x	Input mode
1 0	Push-pull output
1 1	Alternative function: ADC0 input

.5 .4	
0 x	Input mode
1 0	Push-pull output
1 1	Alternative function: ADC1 input

.3 .2	
0 0	Input mode
0 1	Alternative function: BUZ output
1 0	Push-pull output
1 1	Alternative function: ADC2 input

.1 .0	
0 0	Input mode
0 1	Alternative function: TBPWM output
1 0	Push-pull output
1 1	Alternative function: ADC3 input

NOTE: You must be care of the pull-up resistor option.

Figure 9-5. Port 1 Low-Byte Control Register (P1CONL)

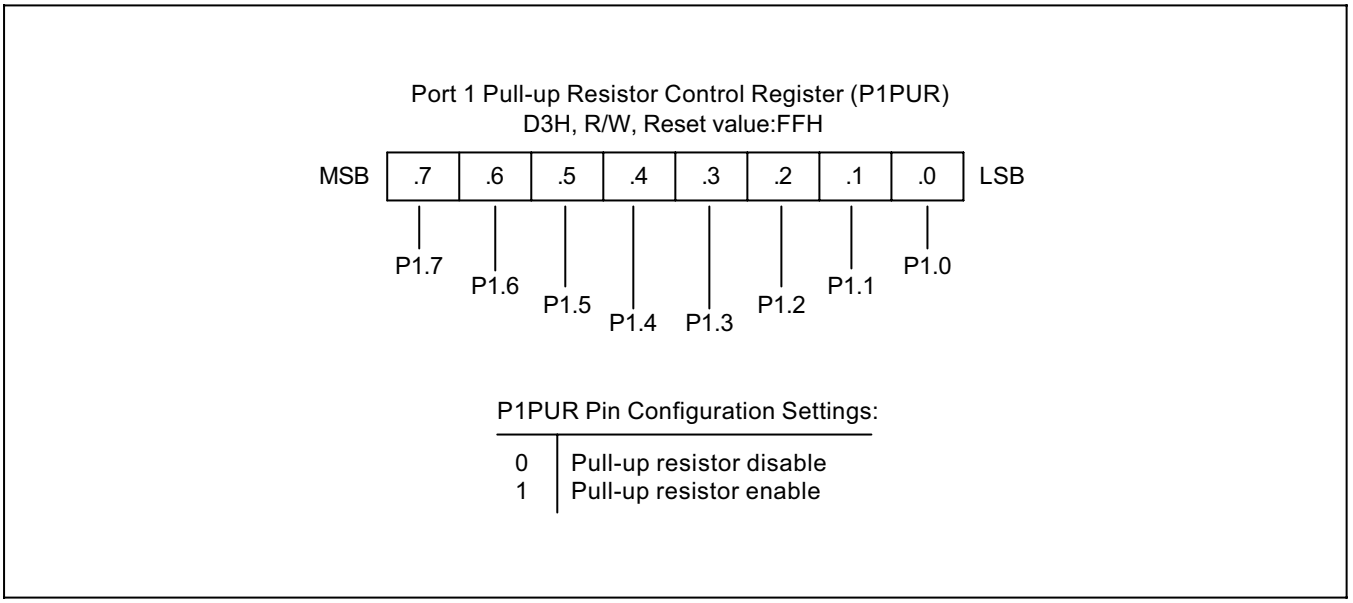


Figure 9-6. Port 1 Pull-up Resistor Control Register (P1PUR)

PORT 2

Port 2 is an 8-bit I/O port that you can use two ways:

— General-purpose I/O

— Alternative function

Port 2 is accessed directly by writing or reading the port 2 data register, P2 at location E2H.

Port 2 Control Register (P2CONH, P2CONL)

Port 2 pins are configured individually by bit-pair settings in two control registers located :
P2CONL (low byte, EBH) and P2CONH (high byte, EAH).

When you select output mode, a push-pull circuit is configured. In input mode, many different selections are available:

— input mode

— Push-pull output mode

— Alternative function: LCD ‘SEG’ signal output – SEG3, SEG4, SEG5, SEG6, SEG7, SEG8, SEG9, SEG10

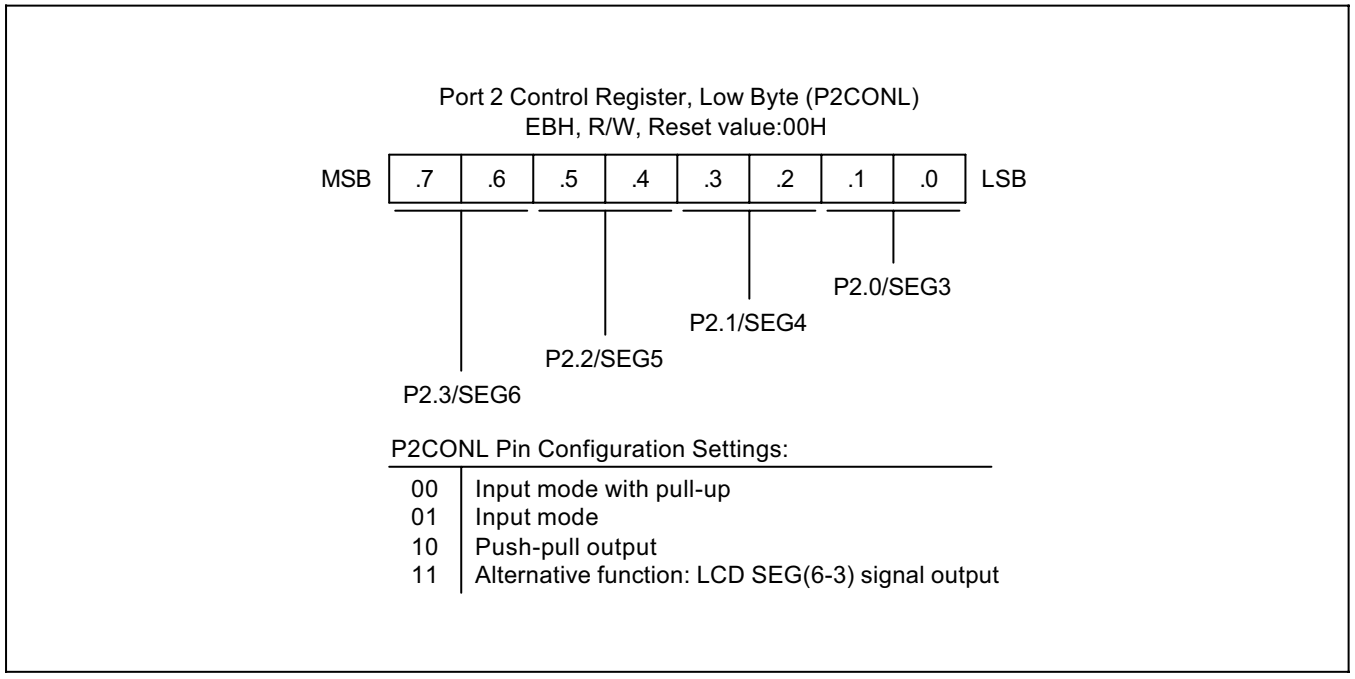
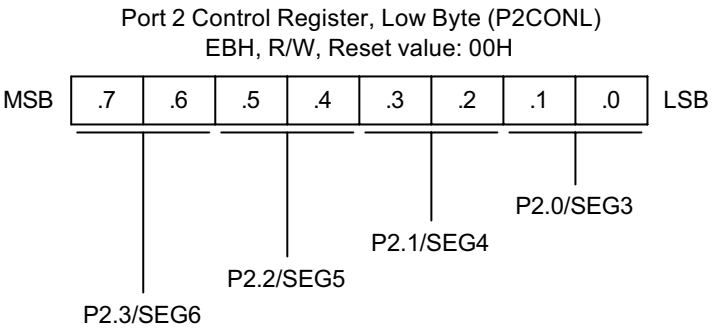


Figure 9-7. Port 2 High-Byte Control Register (P2CONH)



P2CONL Pin Configuration Settings:

0 0	Input mode with pull-up
0 1	Input mode
1 0	Push-pull output
1 1	Alternative function: LCD SEG(6-3) signal output

Figure 9-8. Port 2 Low-Byte Control Register (P2CONL)

PORT 3

Port 3 is an 7-bit I/O Port that you can use two ways:

— General-purpose I/O

— Alternative function

Port 3 is accessed directly by writing or reading the port 3 data register, P3 at location E3H.

Port 3 Control / Interrupt Control Register (P3CONH, P3CONL)

Port 3 pins are configured individually by bit-pair settings in two control registers located:
P3CONL (low byte, EDH) , P3CONH (high byte, ECH).

When you select output mode, a push-pull circuit is configured. In input mode, many different selections are available:

— Input mode.

— Push-pull output mode

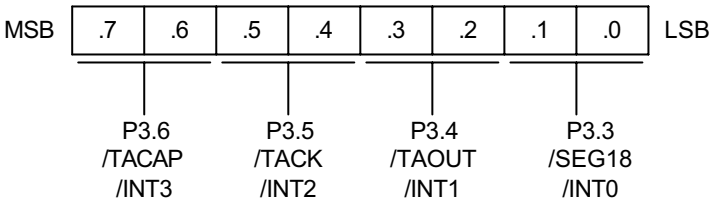
— Alternative function: Timer A signal in/out mode – TAOUT(TAPWM), TACAP, TACK

— Alternative function: External interrupt input – INT0, INT1, INT2, INT3

— Alternative function: LCD ‘SEG’ signal output – SEG15, SEG16, SEG17, SEG18

— Alternative function: UART module – TXD/RXD

Port 3 Control Register, High-Byte (P3CONH)
ECH, R/W, Reset value:00H



.7 .6	
0 0	Input mode with pull-up; External interrupt input (INT3); TACAP
0 1	Input mode; External interrupt input (INT3); TACAP
1 0	Push-pull output
1 1	Open-drain output

.5 .4	
0 0	Input mode with pull-up; External interrupt input (INT2); TACK
0 1	Input mode; External interrupt input (INT2); TACK
1 0	Push-pull output
1 1	Open-drain output

.3 .2	
0 0	Input mode with pull-up; External interrupt input (INT1)
0 1	Input mode; External interrupt input (INT1)
1 0	Push-pull output
1 1	Alternative mode; TAOUT(TAPWM) output

.1 .0	
0 0	Input mode with pull-up; External interrupt input (INT0)
0 1	Input mode; External interrupt input (INT0)
1 0	Push-pull output
1 1	Alternative mode: LCD SEG18 signal output

NOTE: Reset value of P3CONH is determined by Smart Option 3DH .

Figure 9-9. Port 3 High-Byte Control Register (P3CONH)

Port 3 Control Register, Low Byte (P3CONL)
EDH, R/W, Reset value:00H

MSB	.7	.6	.5	.4	.3	.2	.1	.0	LSB

P3.0/SEG15
P3.1/SEG16/RXD
P3.2/SEG17/TXD

.7 .6 .5	
0 0 0	Input mode with pull-up
0 0 1	Input mode
0 1 0	Push-pull output
0 1 1	Alternative mode; TXD output
1 x x	Alternative mode; LCD SEG17 signal output

.4 .3 .2	
0 0 0	Input mode with pull-up; RXD input
0 0 1	Input mode; RXD input
0 1 0	Push-pull output
0 1 1	Alternative mode: RXD output
1 x x	Alternative mode: LCD SEG16 signal output

.1 .0	
0 0	Input mode with pull-up
0 1	Input mode
1 0	Push-pull output
1 1	Alternative mode; LCD SEG15 signal output

Figure 9-10. Port 3 Low-Byte Control Register (P3CONL)

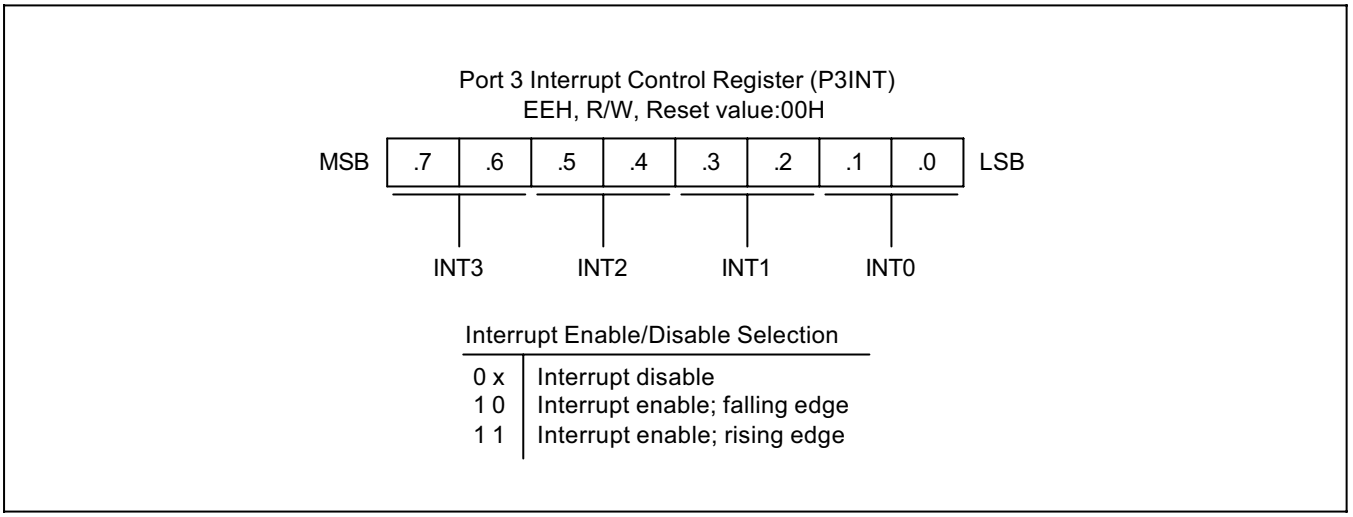


Figure 9-12. Port 3 Interrupt Control Register (P3INT)

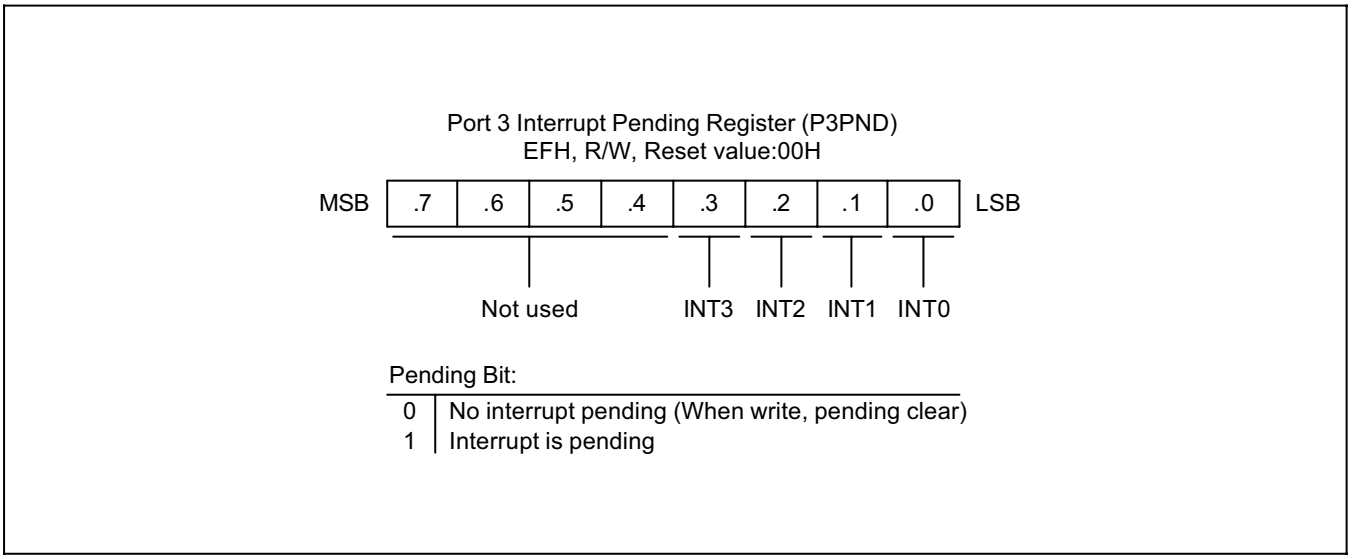


Figure 9-13. Port 3 Interrupt Pending Register (P3PND)

PORT 4

Port 4 is an 7-bit I/O port with individually configurable pins. Port 4 pins are accessed directly by writing or reading the port 4 data register, P4 at location E4H. P4.0-P4.6 can serve as inputs (with or without pull-up), and push-pull output. And they can serve as segment pins for LCD.

Port 4 Control Register (P4CONH, P4CONL)

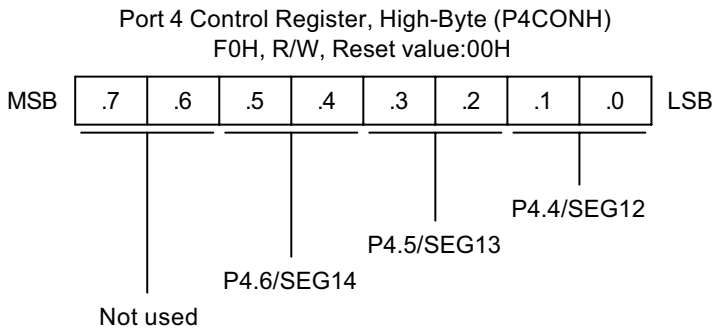
Port 4 pins are configured individually by bit-pair settings in two control registers located :
P4CONL (low byte, F1H) , P4CONH (high byte, F0H)

When you select output mode, a push-pull circuit is configured. In input mode, many different selections are available:

— Input mode.

— Push-pull output mode

— Alternative function: LCD ‘SEG’ signal output – SEG0, SEG1, SEG2, SEG11, SEG12, SEG13, SEG14



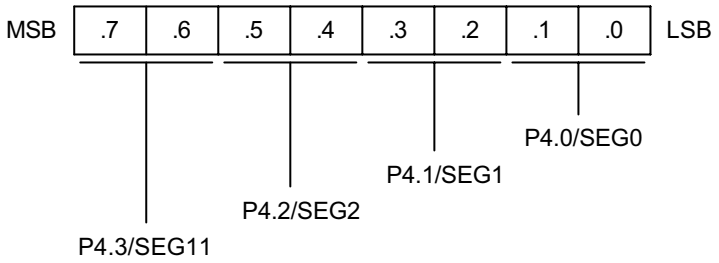
.5 .4	
0 0	Input mode with pull-up
0 1	Input mode
1 0	Push-pull output
1 1	Alternative mode: LCD SEG14 signal output

.3 .2	
0 0	Input mode with pull-up
0 1	Input mode
1 0	Push-pull output
1 1	Alternative mode: LCD SEG13 signal output

.1 .0	
0 0	Input mode with pull-up
0 1	Input mode
1 0	Push-pull output
1 1	Alternative mode: LCD SEG12 signal output

Figure 9-14. Port 4 High-Byte Control Register (P4CONH)

Port 4 Control Register, Low-Byte (P4CONL)
F1H, R/W, Reset value:00H



.7 .6	
0 0	Input mode with pull-up
0 1	Input mode
1 0	Push-pull output
1 1	Alternative mode: LCD SEG11 signal output

.5 .4	
0 0	Input mode with pull-up
0 1	Input mode
1 0	Push-pull output
1 1	Alternative mode: LCD SEG2 signal output

.3 .2	
0 0	Input mode with pull-up
0 1	Input mode
1 0	Push-pull output
1 1	Alternative mode: LCD SEG1 signal output

.1 .0	
0 0	Input mode with pull-up
0 1	Input mode
1 0	Push-pull output
1 1	Alternative mode: LCD SEG0 signal output

Figure 9-15. Port 4 Low-Byte Control Register (P4CONL)

NOTES

10 BASIC TIMER

OVERVIEW

BASIC TIMER (BT)

You can use the basic timer (BT):

— To signal the end of the required oscillation stabilization interval after a reset or a Stop mode release.

The functional components of the basic timer block are:

— Clock frequency divider (fxx divided by 4096, 1024 or 128) with multiplexer
— 8-bit basic timer counter, BTCNT (DDH, read-only)
— Basic timer control register, BTCON (DCH, read/write)

BASIC TIMER CONTROL REGISTER (BTCON)

The basic timer control register, BTCON, is used to select the input clock frequency, to clear the basic timer counter and frequency dividers. It is located in address DCH, and is read/write addressable using register addressing mode.

A reset clears BTCON to '00H'. This enables selects a basic timer clock frequency of $f_{XX}/4096$.

The 8-bit basic timer counter, BTCNT (DDH), can be cleared at any time during normal operation by writing a "1" to BTCON.1. To clear the frequency dividers, write a "1" to BTCON.0.

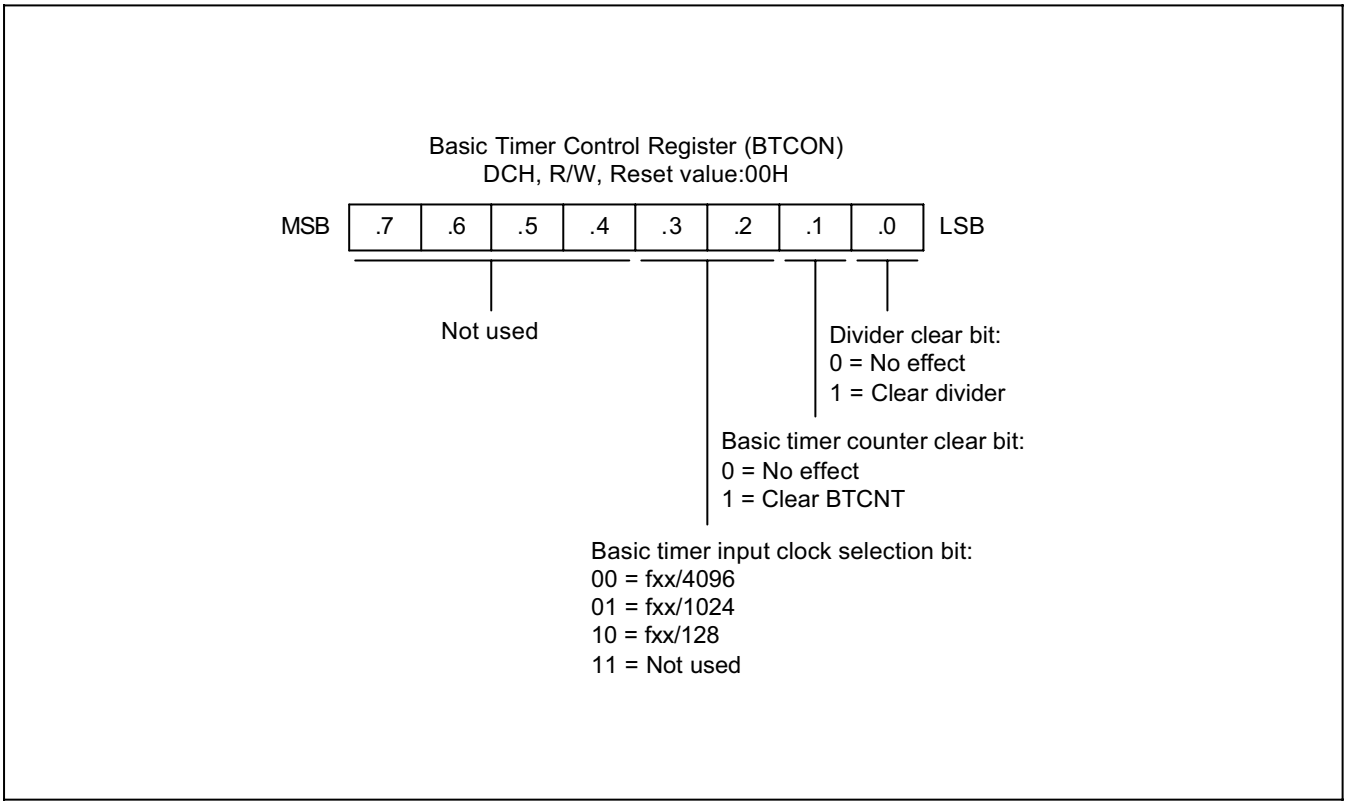


Figure 10-1. Basic Timer Control Register (BTCON)

BASIC TIMER FUNCTION DESCRIPTION

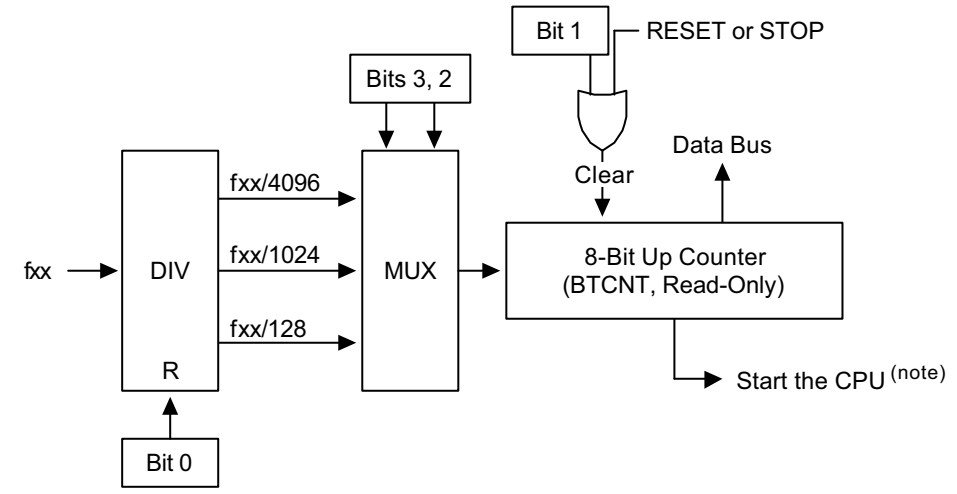
Oscillation Stabilization Interval Timer Function

You can also use the basic timer to program a specific oscillation stabilization interval following a reset or when Stop mode has been released by an external interrupt.

In Stop mode, whenever a reset or an external interrupt occurs, the oscillator starts. The BTCNT value then starts increasing at the rate of fxx/4096 (for reset), or at the rate of the preset clock source (for an external interrupt). When BTCNT.4 overflows, a signal is generated to indicate that the stabilization interval has elapsed and to gate the clock signal off to the CPU so that it can resume normal operation.

In summary, the following events occur when stop mode is released:

1. During stop mode, a power-on reset or an interrupt occurs to trigger the Stop mode release and oscillation starts.
2. If a power-on reset occurred, the basic timer counter will increase at the rate of fxx/4096. If an interrupt is used to release stop mode, the BTCNT value increases at the rate of the preset clock source.
3. Clock oscillation stabilization interval begins and continues until bit 4 of the basic timer counter overflows.
4. When a BTCNT.4 overflow occurs, normal CPU operation resumes.



NOTE: During a power-on reset operation, the CPU is idle during the required oscillation stabilization interval (until bit 4 of the basic timer counter overflows).

Figure 10-2. Basic Timer Block Diagram

NOTES

11 8-BIT TIMER A/B

8-BIT TIMER A

OVERVIEW

The 8-bit timer A is an 8-bit general-purpose timer/counter. Timer A has three operating modes, you can select one of them using the appropriate TACON setting:

— Interval timer mode (Toggle output at TAOUT pin)
— Capture input mode with a rising or falling edge trigger at the TACAP pin
— PWM mode (TAOUT)

Timer A has the following functional components:

— Clock frequency divider (fxx divided by 1024, 256, or 64) with multiplexer
— External clock input pin (TACK)
— 8-bit counter (TACNT), 8-bit comparator, and 8-bit reference data register (TADATA)
— I/O pins for capture input (TACAP) or PWM or match output (TAOUT)
— Timer A overflow interrupt and match/capture interrupt generation
— Timer A control register, TACON (F3H, read/write)

FUNCTION DESCRIPTION

Timer A Interrupts

The timer A module can generate two interrupts: the timer A overflow interrupt (TAOVF), and the timer A match/ capture interrupt (TAINT).

Timer A overflow interrupt pending condition must be cleared by software when it has been serviced. Timer A match/capture interrupt, TAINT pending condition is also cleared by software when it has been serviced.

Interval Timer Function

The timer A module can generate an interrupt: the timer A match interrupt (TAINT).

When timer A interrupt occurs and is serviced by the CPU, the pending condition have to be cleared by software.

In interval timer mode, a match signal is generated and TAOUT is toggled when the counter value is identical to the value written to the TA reference data register, TADATA. The match signal generates a timer A match interrupt and clears the counter.

If, for example, you write the value 10H to TADATA and 0AH to TACON, the counter will increment until it reaches 10H. At this point, the TA interrupt request is generated, the counter value is reset, and counting resumes.

Pulse Width Modulation Mode

Pulse width modulation (PWM) mode lets you program the width (duration) of the pulse that is output at the TAOUT pin. As in interval timer mode, a match signal is generated when the counter value is identical to the value written to the timer A data register. In PWM mode, however, the match signal does not clear the counter. Instead, it runs continuously, overflowing at FFH, and then continues incrementing from 00H.

Although you can use the match signal to generate a timer A overflow interrupt, interrupts are not typically used in PWM-type applications. Instead, the pulse at the TAOUT pin is held to Low level as long as the reference data value is *less than or equal to* (≤) the counter value and then the pulse is held to High level for as long as the data value is *greater than* (>) the counter value. One pulse width is equal to $t_{CLK} \cdot 256$.

Capture Mode

In capture mode, a signal edge that is detected at the TACAP pin opens a gate and loads the current counter value into the TADATA register. You can select rising or falling edges to trigger this operation.

Timer A also gives you capture input source: the signal edge at the TACAP pin. You select the capture input by setting the value of the timer A capture input selection bit in the port 3 high–byte control register, P3CONH, (ECH). When P3CONH.5.4 is 00 and 01, the TACAP input or normal input is selected. When P3CONH.5.4 is set to 10 and 11, output is selected.

Both kinds of timer A interrupts can be used in capture mode: the timer A overflow interrupt is generated whenever a counter overflow occurs; the timer A match/capture interrupt is generated whenever the counter value is loaded into the TADATA register.

By reading the captured data value in TADATA, and assuming a specific value for the timer A clock frequency, you can calculate the pulse width (duration) of the signal that is being input at the TACAP pin.

TIMER A CONTROL REGISTER (TACON)

You use the timer A control register, TACON

— Select the timer A operating mode (interval timer, capture mode and PWM mode)

— Select the timer A input clock frequency

— Clear the timer A counter, TACNT

— Enable the timer A overflow interrupt or timer A match/capture interrupt

— Timer A start/stop

— Clear timer A match/capture interrupt pending conditions

TACON is located at address F3H, and is read/write addressable using Register addressing mode.

A reset clears TACON to '00H'. This sets timer A to normal interval timer mode, selects an input clock frequency of fxx/1024, and disables all timer A interrupts. You can clear the timer A counter at any time during normal operation by writing a "1" to TACON.3. You can start the timer A counter by writing a “1” to TACON.0.

The timer A overflow interrupt (TAOVF) has the vector address 00H-01H. When a timer A overflow interrupt occurs and is serviced by the CPU, but the pending condition must clear by software.

To enable the timer A match/capture interrupt , you must write TACON.1 to "1". To generate the exact time interval, you should write TACON.3 and .0, which cleared counter and interrupt pending bit. When interrupt service routine is served, the pending condition must be cleared by software by writing a ‘0’ to the interrupt pending bit.

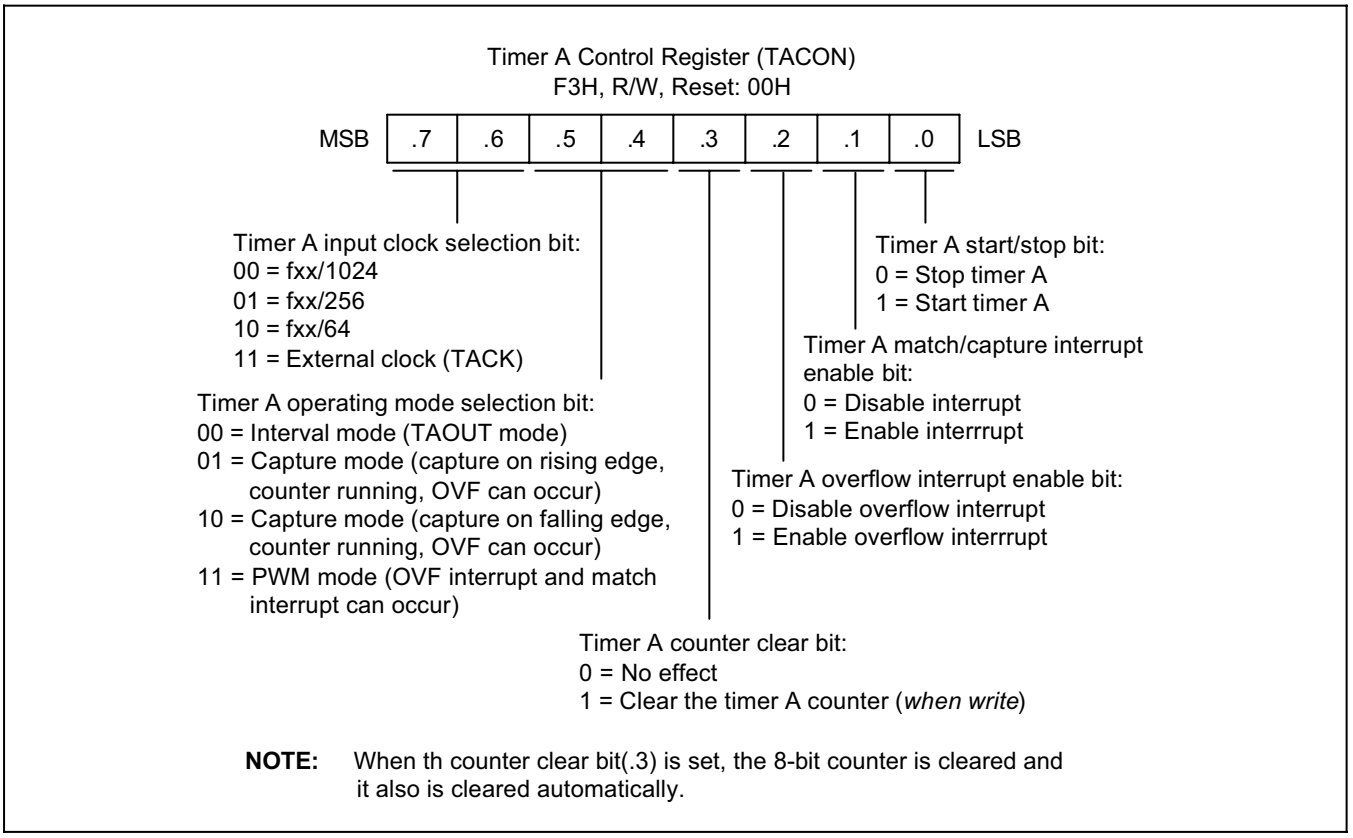

Timer A Control Register (TACON)
F3H, R/W, Reset: 00H
MSB
.7
.6
.5
.4
.3
.2
.1
.0
LSB
Timer A input clock selection bit:
00 = fxx/1024
01 = fxx/256
10 = fxx/64
11 = External clock (TACK)
Timer A operating mode selection bit:
00 = Interval mode (TAOUT mode)
01 = Capture mode (capture on rising edge, counter running, OVF can occur)
10 = Capture mode (capture on falling edge, counter running, OVF can occur)
11 = PWM mode (OVF interrupt and match interrupt can occur)
Timer A counter clear bit:
0 = No effect
1 = Clear the timer A counter (when write)
Timer A overflow interrupt enable bit:
0 = Disable overflow interrupt
1 = Enable overflow interrrupt
Timer A match/capture interrupt enable bit:
0 = Disable interrupt
1 = Enable interrrupt
Timer A start/stop bit:
0 = Stop timer A
1 = Start timer A
NOTE: When th counter clear bit(.3) is set, the 8-bit counter is cleared and it also is cleared automatically.

Figure 11-1. Timer A Control Register (TACON)

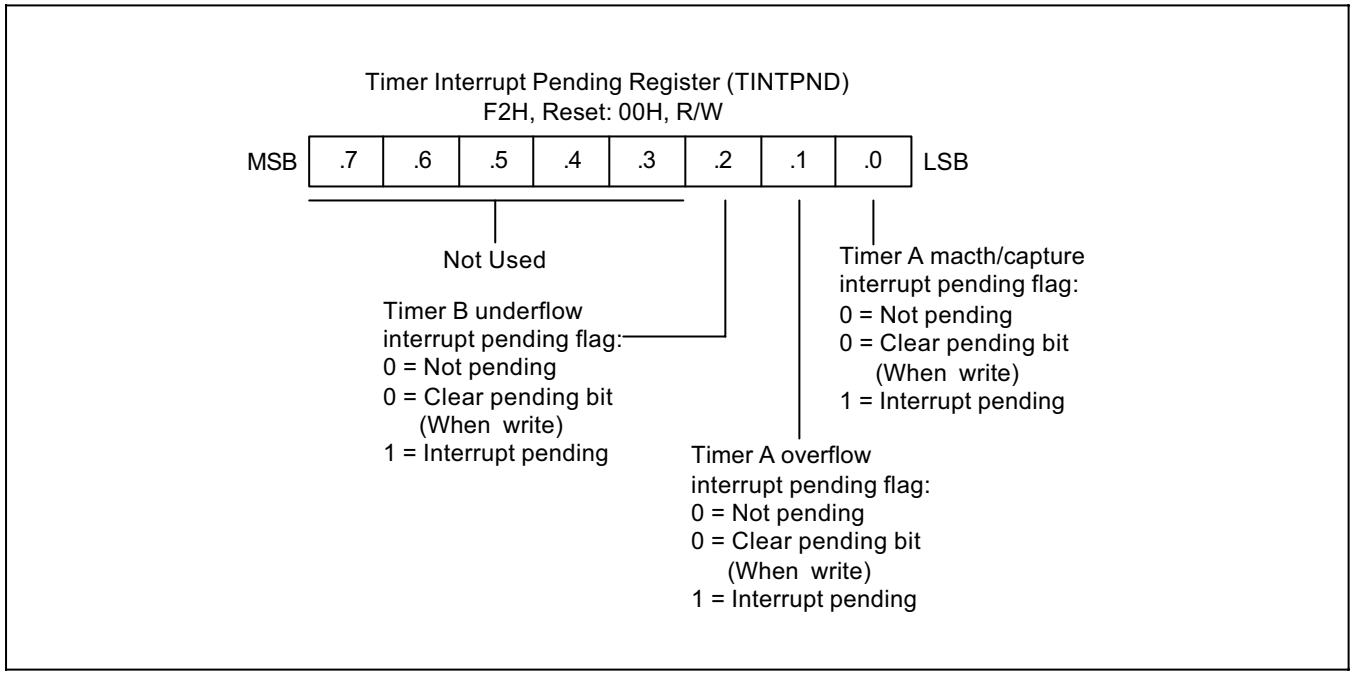


Figure 11-2. Timer interrupts Pending Register (TINTPND)

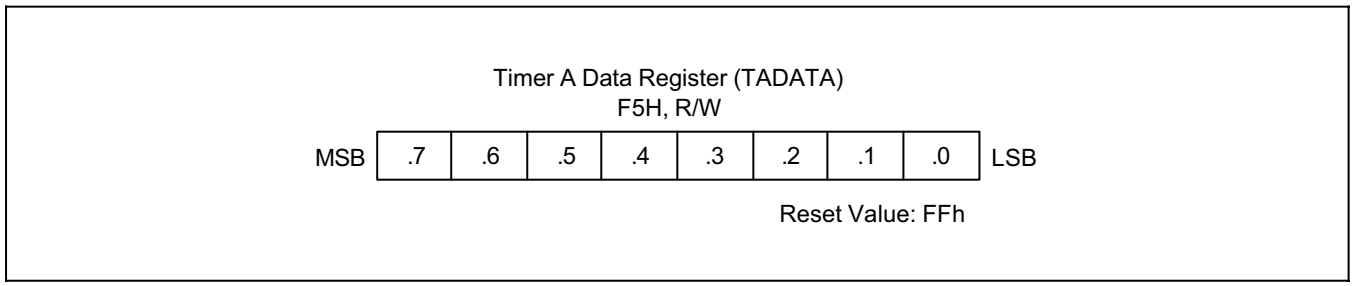
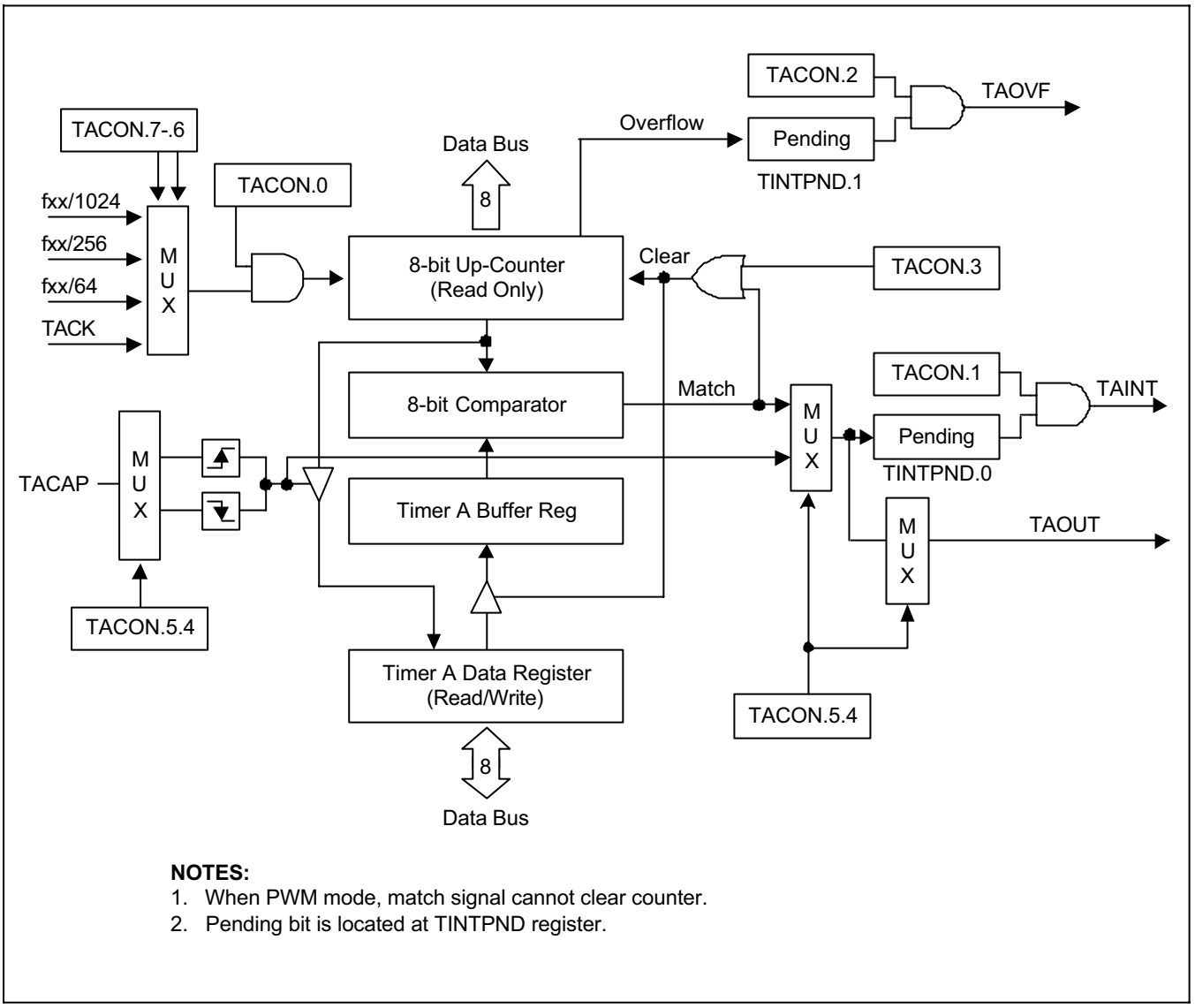


Figure 11-3. Timer A DATA Register (TADATA)

BLOCK DIAGRAM



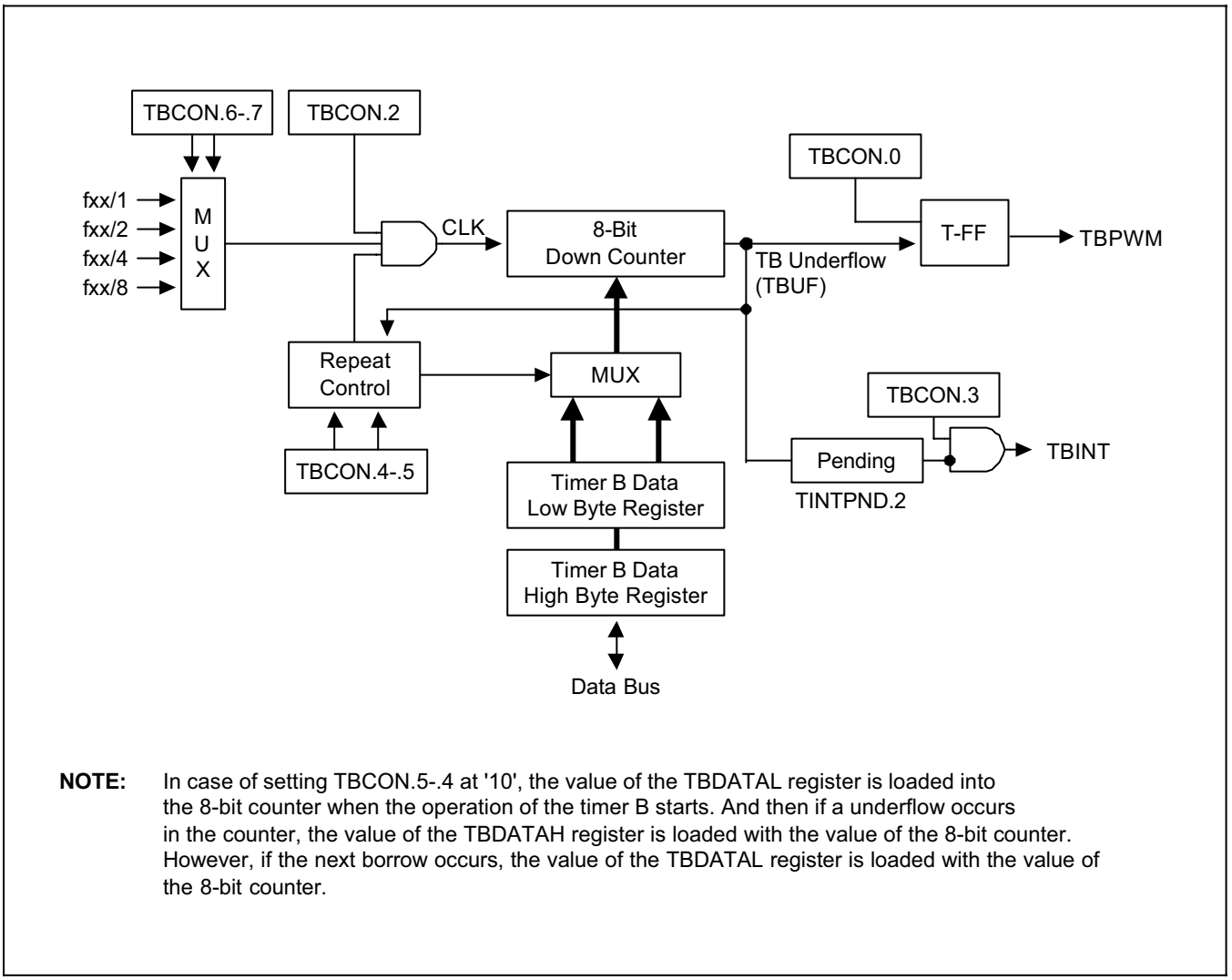
NOTES:
1. When PWM mode, match signal cannot clear counter.
2. Pending bit is located at TINTPND register.

Figure 11-4. Timer A Functional Block Diagram

8-BIT TIMER B

OVERVIEW

The S3C9484/C9488/F9488 micro-controller has an 8-bit counter called timer B. Timer B, which can be used to generate the carrier frequency of a remote controller signal. As a normal interval timer, generating a timer B interrupt at programmed time intervals.



NOTE: In case of setting TBCON.5-.4 at '10', the value of the TBDATAL register is loaded into the 8-bit counter when the operation of the timer B starts. And then if a underflow occurs in the counter, the value of the TBDATAH register is loaded with the value of the 8-bit counter. However, if the next borrow occurs, the value of the TBDATAL register is loaded with the value of the 8-bit counter.

Figure 11-5. Timer B Functional Block Diagram

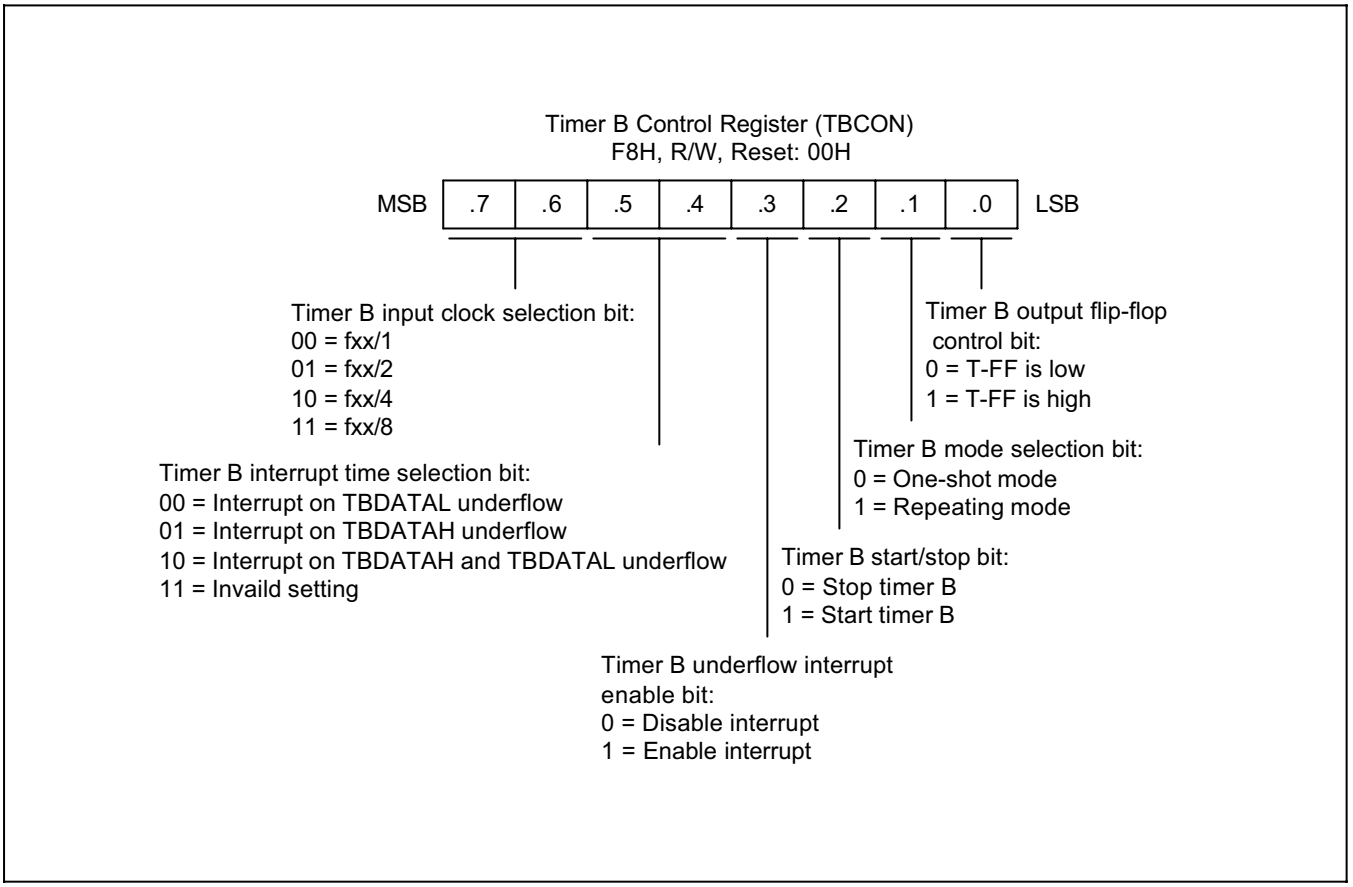


Figure 11-6. Timer B Control Register (TBCON)

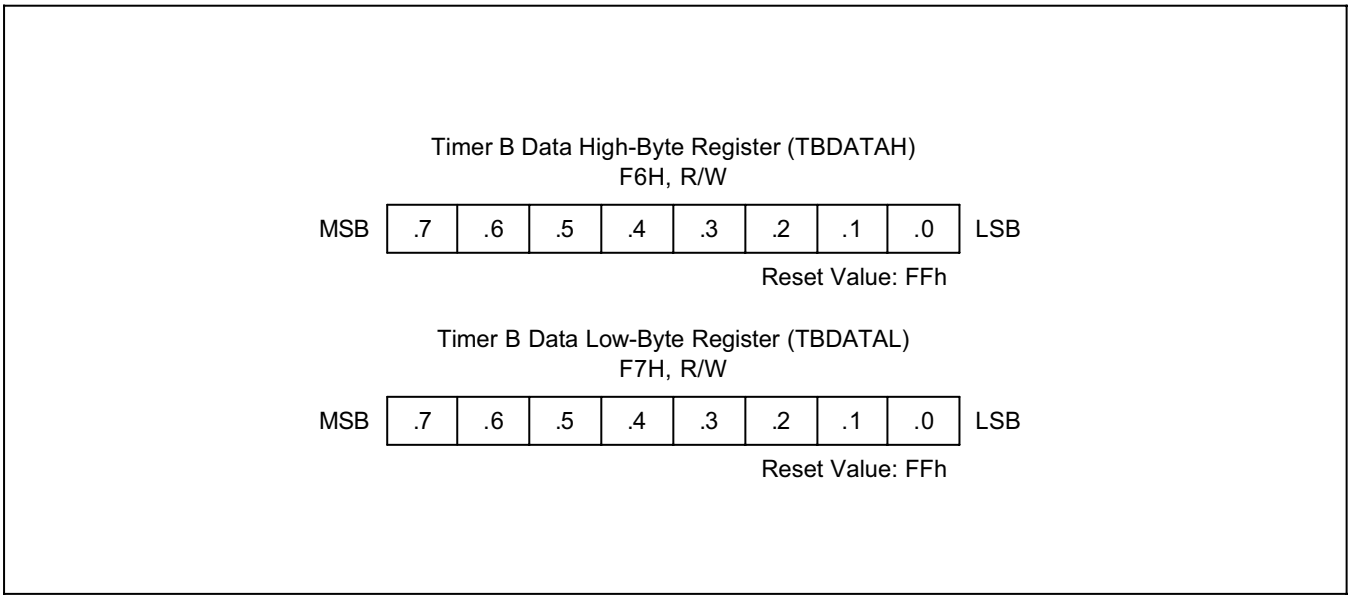


Figure 11-7. Timer B DATA Registers (TBDATAH, TBDATAL)

+ Programming Tip – Using Timer A (fxx – 8MHz, 800µsec interval)

```
            .INCLUDE    "C:\SKSTUDIO\INCLUDE\REG\S3C9488.REG"

            VECTOR      00H,F9488_INT

            .ORG        003CH
             DB         0FFH
             DB         0FFH
             DB         01100000B          ;DISABLE LVR
             DB         00000011B          ;SUB OSCILLATOR,BT OVERFLOW, RESET PIN ENALBE

            .ORG        100H

RESET:
            DI
            LD          WDTCON,#10101010B
            LD          BTCON,#0001011B
            LD          CLKCON,#00011000B
            LD          SP,#0C0H
            LD          SYM,#00H
            LD          OSCCON,#00000000B
            LD          P3CONH,#10101110B  ;TAOUT
            LD          TADATA,#100
            LD          TACON,#10001011B   ;Fxx/64,INTERVAL MODE,TIMER START.
            EI
;=============================================================================

MAIN
            JP          MAIN
;=============================================================================
F9488_INT
            TM          TINTPND,#01H       ;CHECK WHAT INTERRUPT IS ENABLED
            JP          NC,TA_MC_INT

            ;.....

            IRET

TA_MC_INT
            LD          TINTPND,#0
            NOP
            NOP
            IRET

.END
```

+ Programming Tip – Using Timer B (fxx – 8MHz, Duty – 2:8, 80kHz)

```
        .INCLUDE        "C:\SKSTUDIO\INCLUDE\REG\S3C9488.REG"

        VECTOR          00H,F9488_INT

        .ORG    003CH

            DB          0FFH
            DB          0FFH
            DB          01100000B           ;DISABLE LVR
            DB          00000011B           ;SUB OSCILLATOR,BT OVERFLOW, RESET PIN ENALBE

        .ORG            100H

RESET:

            DI
            LD          WDTCON,#10101010B
            LD          BTCON,#0001011B
            LD          CLKCON,#00011000B
            LD          SP,#0C0H
            LD          SYM,#00H
            LD          OSCCON,#00000000B
            LD          P1CONL,#10101001B   ;TB PWM
            LD          TBDATAH,#79
            LD          TBDATAL,#19
            LD          TBCON,#00101111B    ;Fxx,REPEAT MODE,FLIP-FLOP HIGH,TIMER START.
            EI

;=============================================================================
MAIN
            JP          MAIN
;=============================================================================
F9488_INT
            TM          TINTPND,#04H        ;CHECK WHAT INTERRUPT IS ENABLED
            JP          NC,TB_UF_INT

            ;.....

            IRET


TB_UF_INT
            LD          TINTPND,#0
            NOP
            NOP
            IRET
.END
```

NOTES

12 UART

OVERVIEW

The UART block has a full-duplex serial port with programmable operating modes: There is one synchronous mode and three UART (Universal Asynchronous Receiver/Transmitter) modes:

— Shift Register I/O with baud rate of fxx/(16 × (16bit BRDATA+1))

— 8-bit UART mode; variable baud rate, fxx/(16 × (16bit BRDATA+1))

— 9-bit UART mode; variable baud rate, fxx/(16 × (16bit BRDATA+1))

UART receive and transmit buffers are both accessed via the data register, UDATA, is at address FFH. Writing to the UART data register loads the transmit buffer; reading the UART data register accesses a physically separate receive buffer.

When accessing a receive data buffer (shift register), reception of the next byte can begin before the previously received byte has been read from the receive register. However, if the first byte has not been read by the time the next byte has been completely received, the first data byte will be lost (Overrun error).

In all operating modes, transmission is started when any instruction (usually a write operation) uses the UDATA register as its destination address. In mode 0, serial data reception starts when the receive interrupt pending bit (UARTPND.1) is "0" and the receive enable bit (UARTCON.4) is "1". In mode 1 and 2, reception starts whenever an incoming start bit ("0") is received and the receive enable bit (UARTCON.4) is set to "1".

PROGRAMMING PROCEDURE

To program the UART modules, follow these basic steps:

1. Configure P3.1 and P3.2 to alternative function (RXD (P3.1), TXD (P3.2)) for UART module by setting the P3CONL register to appropriate value.
2. Load an 8-bit value to the UARTCON control register to properly configure the UART I/O module.
3. For parity generation and check in UART mode 2, set parity enable bit (UARTPND.5) to “1”.
4. For interrupt generation, set the UART interrupt enable bit (UARTCON.1 or UARTCON.0) to "1".
5. When you transmit data to the UART buffer, write transmit data to UDATA, the shift operation starts.
6. When the shift operation (transmit/receive) is completed, UART pending bit (UARTPND.1 or UARTPND.0) is set to "1" and an UART interrupt request is generated.

UART CONTROL REGISTER (UARTCON)

The control register for the UART is called UARTCON at address FDH. It has the following control functions:

— Operating mode and baud rate selection

— Multiprocessor communication and interrupt control

— Serial receive enable/disable control

— 9th data bit location for transmit and receive operations (mode 2)

— Parity generation and check for transmit and receive operations (mode 2)

— UART transmit and receive interrupt control

A reset clears the UARTCON value to "00H". So, if you want to use UART module, you must write appropriate value to UARTCON.

UART Control Register (UARTCON)
FDH, R/W, Reset Value: 00H

MSB	MS1	MS0	MCE	RE	TB8	RB8	RIE	TIE	LSB

Operating mode and baud rate selection bits (see table below)

Multiprocessor communication enable bit (mode 2 only):(1)
0 = Disable
1 = Enable

Serial data receive enable bit:
0 = Disable
1 = Enable

If parity disable mode (PEN = 0), location of the 9th data bit to be transmitted in UART mode 2 ("0" or "1").

If parity enable mode (PEN = 1), Even/odd parity selection bit for transmit data in UART mode 2;
0: Even parity bit generation for transmit data
1: Odd parity bit generation for transmit data

Transmit interrupt enable bit:
0 = Disable
1 = Enable

Received interrupt enable bit:
0 = Disable
1 = Enable

If parity disable mode (PEN = 0), location of the 9th data bit that was received in UART mode 2 ("0" or "1").

If parity enable mode (PEN = 1), Even/odd parity selection bit for receive data in UART mode 2.
0: Even parity check for the received data
1: Odd parity check for the received data

MS1	MS0	Mode	Description	Baud Rate
0	0	0	Shift register	fxx / (16 x (16bit BRDATA + 1))
0	1	1	8-bit UART	fxx / (16 x (16bit BRDATA + 1))
1	x	2	9-bit UART	fxx / (16 x (16bit BRDATA + 1))

NOTES:
1. In mode 2, if the UARTCON.5 bit is set to "1" then the receive interrupt will not be activated if the received 9th data bit is "0". In mode 1, if UARTCON.5 = "1" then the receive interrut will not be activated if a valid stop bit was not received.
2. The descriptions for 8-bit and 9-bit UART mode do not include start and stop bits of serial data for receiving and transmitting.
3. Parity enable bits, PEN, is located in the UARTPND register at address FEH.
4. Parity enable and parity error check can be available in 9-bit UART mode (Mode 2) only.

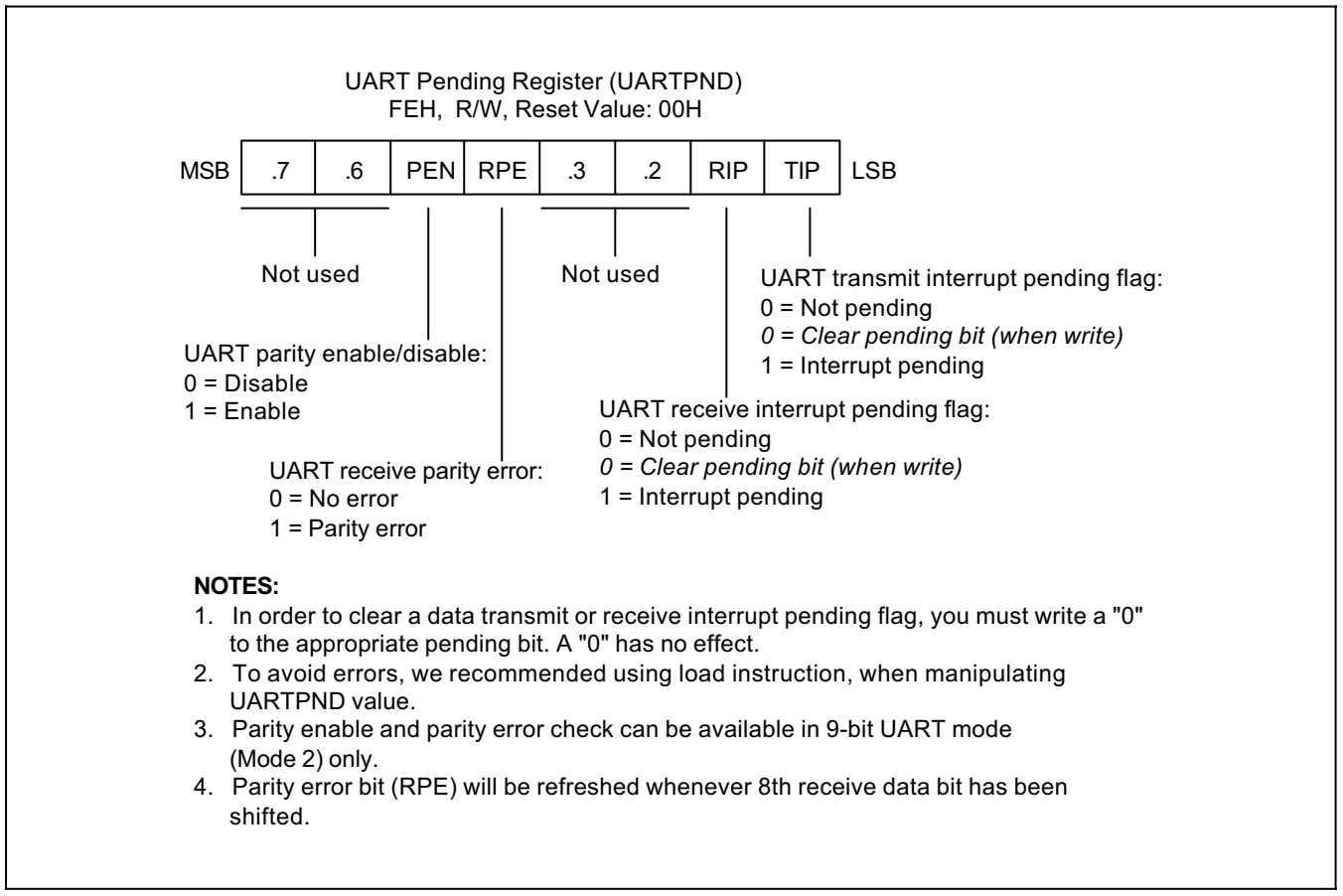
Figure 12-1. UART Control Register (UARTCON)

UART INTERRUPT PENDING REGISTER (UARTPND)

The UART interrupt pending register, UARTPND is located at address FEH. It contains the UART data transmit interrupt pending bit (UARTPND.0) and the receive interrupt pending bit (UARTPND.1).

In mode 0 of the UART module, the receive interrupt pending flag UARTPND.1 is set to "1" when the 8th receive data bit has been shifted. In mode 1 or 2, the UARTPND.1 bit is set to "1" at the halfway point of the stop bit's shift time. When the CPU has acknowledged the receive interrupt pending condition, the UARTPND.1 flag must be cleared by software in the interrupt service routine.

In mode 0 of the UART module, the transmit interrupt pending flag UARTPND.0 is set to "1" when the 8th transmit data bit has been shifted. In mode 1 or 2, the UARTPND.0 bit is set at the start of the stop bit. When the CPU has acknowledged the transmit interrupt pending condition, the UARTPND.0 flag must be cleared by software in the interrupt service routine.



NOTES:

1. In order to clear a data transmit or receive interrupt pending flag, you must write a "0" to the appropriate pending bit. A "0" has no effect.
2. To avoid errors, we recommended using load instruction, when manipulating UARTPND value.
3. Parity enable and parity error check can be available in 9-bit UART mode (Mode 2) only.
4. Parity error bit (RPE) will be refreshed whenever 8th receive data bit has been shifted.

Figure 12-2. UART Interrupt Pending Register (UARTPND)

In mode 2 (9-bit UART data), by setting the parity enable bit (PEN) of UARTPND register to '1', the 9th data bit of transmit data will be an automatically generated parity bit. Also, the 9th data bit of the received data will be treated as a parity bit for checking the received data.

In parity enable mode (PEN = 1), UARTCON.3 (TB8) and UARTCON.2 (RB8) will be a parity selection bit for transmit and receive data respectively. The UARTCON.3 (TB8) is for settings of the even parity generation (TB8 = 0) or the odd parity generation (TB8 = 0) in the transmit mode. The UARTCON.2 (RB8) is also for settings of the even parity checking (RB8= 0) or the odd parity checking (RB8 =1) in the receive mode. The parity enable (generation/checking) functions are not available in UART mode 0 and 1.

If you don't want to use a parity mode, UARTCON.2 (RB8) and UARTCON.3 (TB8) are a normal control bit as the 9th data bit, in this case, PEN must be disable ("0") in mode 2. Also it is needed to select the 9th data bit to be transmitted by writing TB8 to "0" or "1".

The receive parity error flag (RPE) will be set to '0' or '1' depending on parity error whenever the 8th data bit of the receive data has been shifted.

UART DATA REGISTER (UDATA)

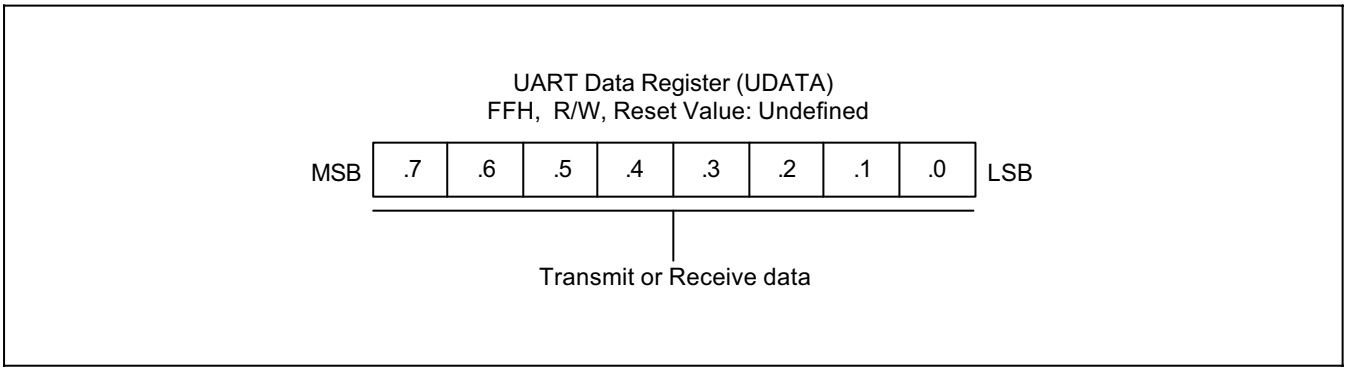


Figure 12-3. UART Data Register (UDATA)

UART BAUD RATE DATA REGISTER (BRDATAH, BRDATAL)

The value stored in the UART baud rate register, (BRDATAH, BRDATAL), lets you determine the UART clock rate (baud rate).

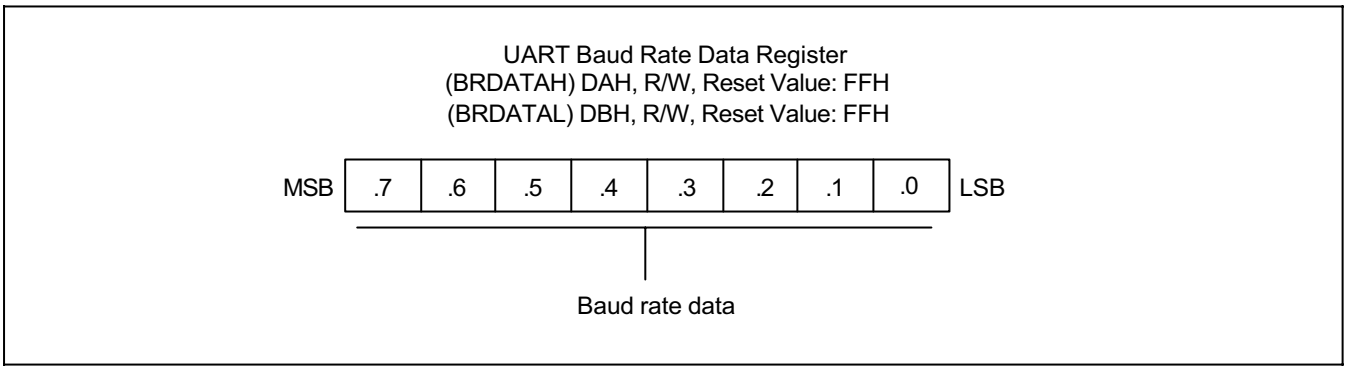


Figure 12-4. UART Baud Rate Data Register (BRDATAH, BRDATAL)

BAUD RATE CALCULATIONS

The baud rate is determined by the baud rate data register, 16bit BRDATA

Mode 0 baud rate = fxx/(16 × (16Bit BRDATA + 1))
Mode 1 baud rate = fxx/(16 × (16Bit BRDATA + 1))
Mode 2 baud rate = fxx/(16 × (16Bit BRDATA + 1))

Table 12-1. Commonly Used Baud Rates Generated by 16-bit BRDATA

Baud Rate	Oscillation Clock	BRDATAH		BRDATAL	
		Decimal	Hex	Decimal	Hex
230,400 Hz	11.0592 MHz	0	0H	02	02H
115,200 Hz	11.0592 MHz	0	0H	05	05H
57,600 Hz	11.0592 MHz	0	0H	11	0BH
38,400 Hz	11.0592 MHz	0	0H	17	11H
19,200 Hz	11.0592 MHz	0	0H	35	23H
9,600 Hz	11.0592 MHz	0	0H	71	47H
4,800 Hz	11.0592 MHz	0	0H	143	8FH
76,800 Hz	10 MHz	0	0H	7	7H
38,400 Hz	10 MHz	0	0H	15	FH
19,200 Hz	10 MHz	0	0H	31	1FH
9,600 Hz	10 MHz	0	0H	64	40H
4,800 Hz	10 MHz	0	0H	129	81H
2,400 Hz	10 MHz	1	1H	3	3H
600 Hz	10 MHz	4	4H	16	10H
38,461 Hz	8 MHz	0	0H	12	0CH
12,500 Hz	8 MHz	0	0H	39	27H
19,230 Hz	4 MHz	0	0H	12	0CH
9,615 Hz	4 MHz	0	0H	25	19H

BLOCK DIAGRAM

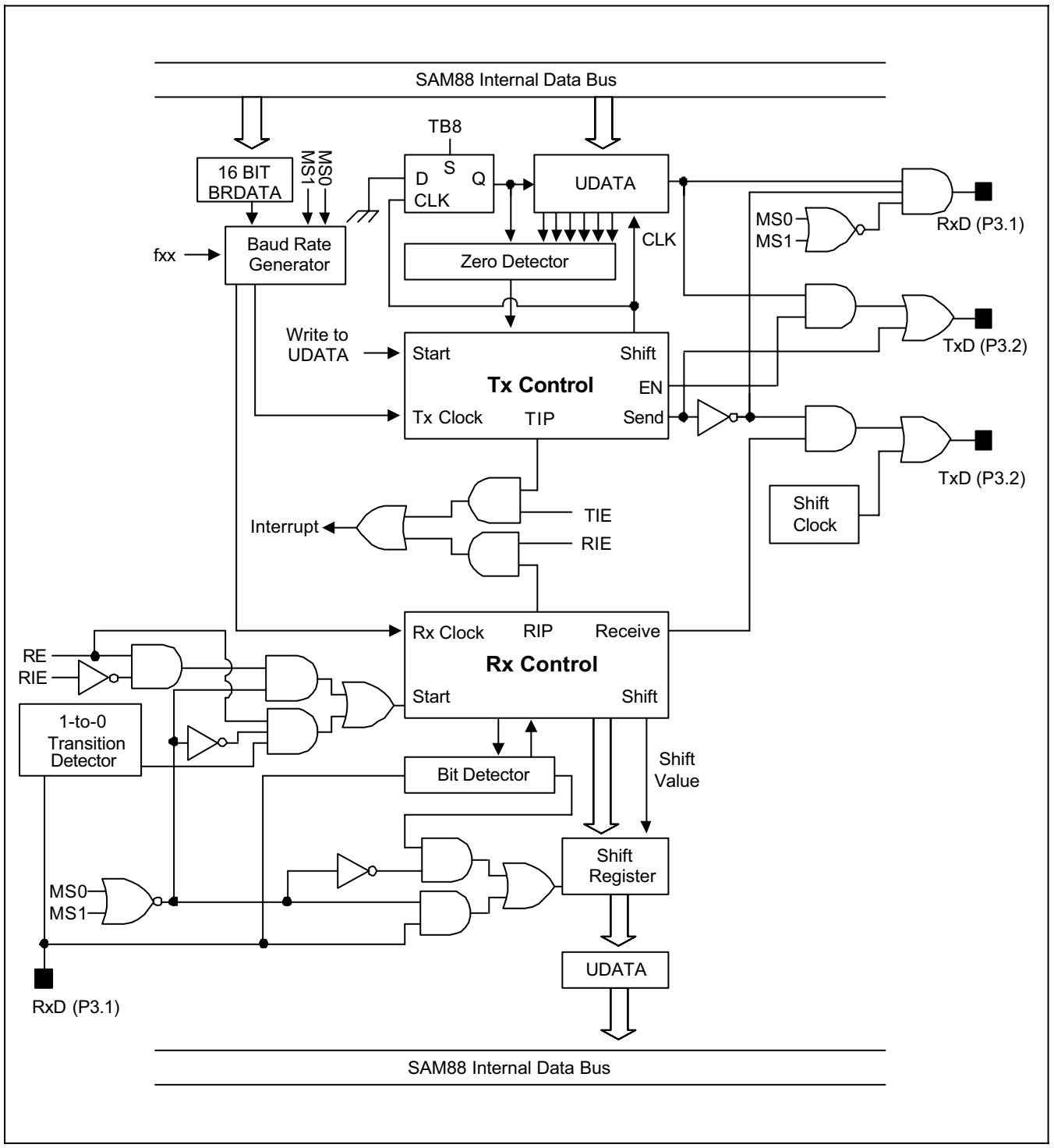


Figure 12-5. UART Functional Block Diagram

UART MODE 0 FUNCTION DESCRIPTION

In mode 0, UART is input and output through the RxD (P3.1) pin and TxD (P3.2) pin outputs the shift clock. Data is transmitted or received in 8-bit units only. The LSB of the 8-bit value is transmitted (or received) first.

Mode 0 Transmit Procedure

1. Select mode 0 by setting UARTCON.6 and .7 to "00B".
2. Write transmission data to the shift register UDATA (FFH) to start the transmission operation.

Mode 0 Receive Procedure

1. Select mode 0 by setting UATCON.6 and .7 to "00B".
2. Clear the receive interrupt pending bit (UARTPND.1) by writing a "0" to UARTPND.1.
3. Set the UART receive enable bit (UARTCON.4) to "1".
4. The shift clock will now be output to the TxD (P3.2) pin and will read the data at the RxD (P3.1) pin. A UART receive interrupt (vector 00H-01H) occurs when UARTCON.1 is set to "1".

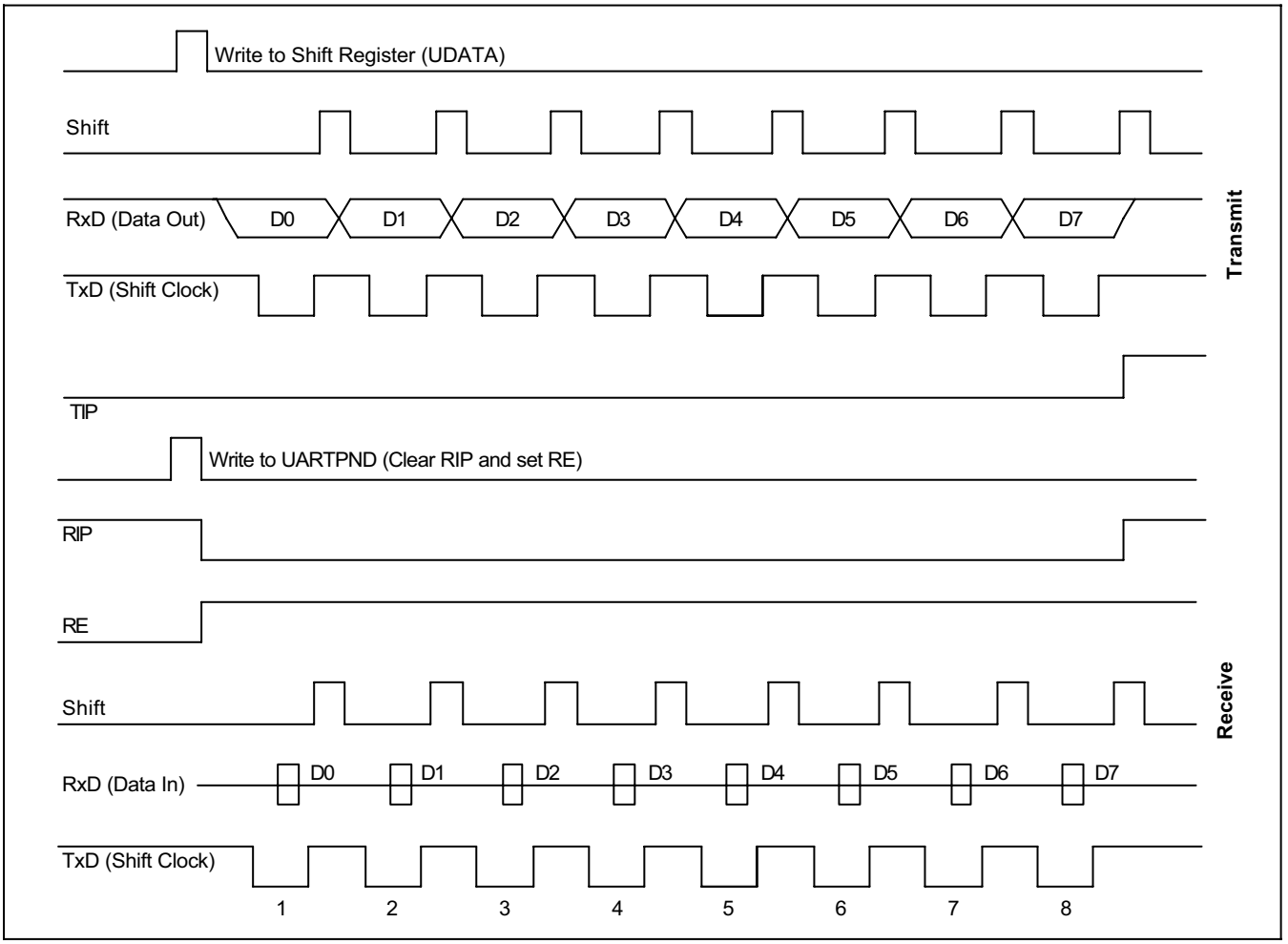


Figure 12-6. Timing Diagram for UART Mode 0 Operation

UART MODE 1 FUNCTION DESCRIPTION

In mode 1, 10-bits are transmitted (through the TxD (P3.2) pin) or received (through the RxD (P3.1) pin). Each data frame has three components:

— Start bit ("0")
— 8 data bits (LSB first)
— Stop bit ("1")

When receiving, the stop bit is written to the RB8 bit in the UARTCON register. The baud rate for mode 1 is variable.

Mode 1 Transmit Procedure

1. Select the baud rate generated by 16bit BRDATA.
2. Select mode 1 (8-bit UART) by setting UARTCON bits 7 and 6 to '01B'.
3. Write transmission data to the shift register UDATA (FFH). The start and stop bits are generated automatically by hardware.

Mode 1 Receive Procedure

1. Select the baud rate to be generated by 16bit BRDATA.
2. Select mode 1 and set the RE (Receive Enable) bit in the UARTCON register to "1".
3. The start bit low ("0") condition at the RxD (P3.1) pin will cause the UART module to start the serial data receive operation.

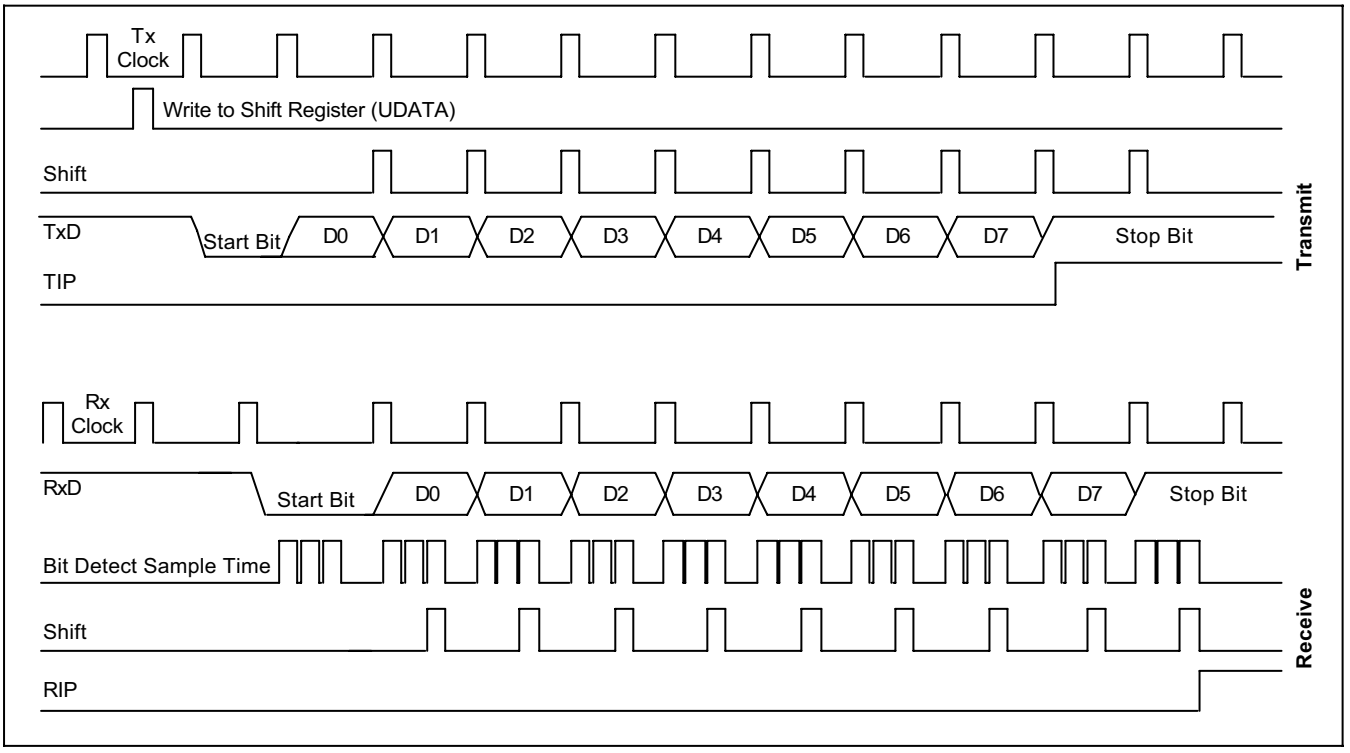


Figure 12-7. Timing Diagram for UART Mode 1 Operation

UART MODE 2 FUNCTION DESCRIPTION

In mode 2, 11-bits are transmitted (through the TxD pin) or received (through the RxD pin). Each data frame has four components:

— Start bit ("0")

— 8 data bits (LSB first)

— Programmable 9th data bit or parity bit

— Stop bit ("1")

< In parity disable mode (PEN = 0) >

The 9th data bit to be transmitted can be assigned a value of "0" or "1" by writing the TB8 bit (UARTCON.3). When receiving, the 9th data bit that is received is written to the RB8 bit (UARTCON.2), while the stop bit is ignored. The baud rate for mode 2 is fosc/(16 x (16bit BRDATA + 1)) clock frequency.

< In parity enable mode (PEN = 1) >

The 9th data bit to be transmitted can be an automatically generated parity of "0" or "1" depending on a parity generation by means of TB8 bit (UARTCON.3). When receiving, the received 9th data bit is treated as a parity for checking receive data by means of the RB8 bit (UARTCON.2), while the stop bit is ignored. The baud rate for mode 2 is fosc/(16 × (16bit BRDATA + 1)) clock frequency.

Mode 2 Transmit Procedure

1. Select the baud rate generated by 16bit BRDATA.
2. Select mode 2 (9-bit UART) by setting UARTCON bits 6 and 7 to '10B'. Also, select the 9th data bit to be transmitted by writing TB8 to "0" or "1" and set PEN bit of UARTPND register to “0” if you don’t use a parity mode. If you want to use the parity enable mode, select the parity bit to be transmitted by writing TB8 to "0" or "1" and set PEN bit of UARTPND register to “1”.
3. Write transmission data to the shift register, UDATA (FFH), to start the transmit operation.

Mode 2 Receive Procedure

1. Select the baud rate to be generated by 16bit BRDATA.
2. Select mode 2 and set the receive enable bit (RE) in the UARTCON register to "1".
3. If you don’t use a parity mode, set PEN bit of UARTPND register to “0” to disable parity mode.
   If you want to use the parity enable mode, select the parity type to be check by writing TB8 to "0" or "1" and set PEN bit of UARTPND register to “1”. Only 8 bits (Bit0 to Bit7) of received data are available for data value.
4. The receive operation starts when the signal at the RxD pin goes to low level.

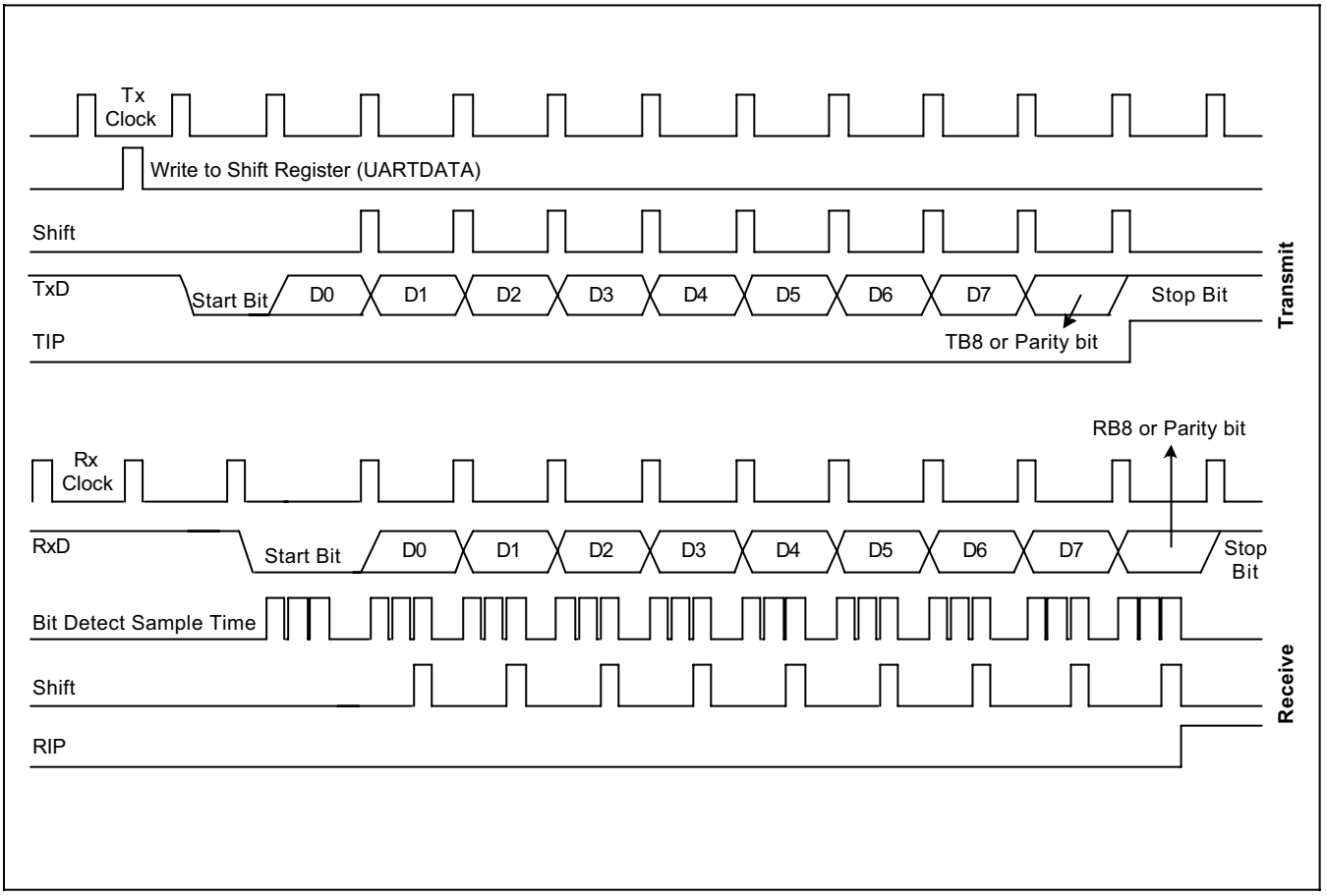


Figure 12-8. Timing Diagram for UART Mode 2 Operation

SERIAL COMMUNICATION FOR MULTIPROCESSOR CONFIGURATIONS

The S3C9-series multiprocessor communication features let a "master" S3C9484/C9488/F9488 send a multiple-frame serial message to a "slave" device in a multi- S3C9484/C9488/F9488 configuration. It does this without interrupting other slave devices that may be on the same serial line.

This feature can be used only in UART mode 2 with the parity disable mode. In mode 2, 9 data bits are received. The 9th bit value is written to RB8 (UARTCON.2). The data receive operation is concluded with a stop bit. You can program this function so that when the stop bit is received, the serial interrupt will be generated only if RB8 = "1".

To enable this feature, you set the MCE bit in the UARTCON registers. When the MCE bit is "1", serial data frames that are received with the 9th bit = "0" do not generate an interrupt. In this case, the 9th bit simply separates the address from the serial data.

Sample Protocol for Master/Slave Interaction

When the master device wants to transmit a block of data to one of several slaves on a serial line, it first sends out an address byte to identify the target slave. Note that in this case, an address byte differs from a data byte: In an address byte, the 9th bit is "1" and in a data byte, it is "0".

The address byte interrupts all slaves so that each slave can examine the received byte and see if it is being addressed. The addressed slave then clears its MCE bit and prepares to receive incoming data bytes.

The MCE bits of slaves that were not addressed remain set, and they continue operating normally while ignoring the incoming data bytes.

While the MCE bit setting has no effect in mode 0, it can be used in mode 1 to check the validity of the stop bit. For mode 1 reception, if MCE is "1", the receive interrupt will be issue unless a valid stop bit is received.

Setup Procedure for Multiprocessor Communications

Follow these steps to configure multiprocessor communications:

1. Set all S3C9484/C9488/F9488 devices (masters and slaves) to UART mode 2 with parity disable.
2. Write the MCE bit of all the slave devices to "1".
3. The master device's transmission protocol is:
   - First byte: the address identifying the target slave device (9th bit = "1")
   - Next bytes: data (9th bit = "0")
4. When the target slave receives the first byte, all of the slaves are interrupted because the 9th data bit is "1". The targeted slave compares the address byte to its own address and then clears its MCE bit in order to receive incoming data. The other slaves continue operating normally.

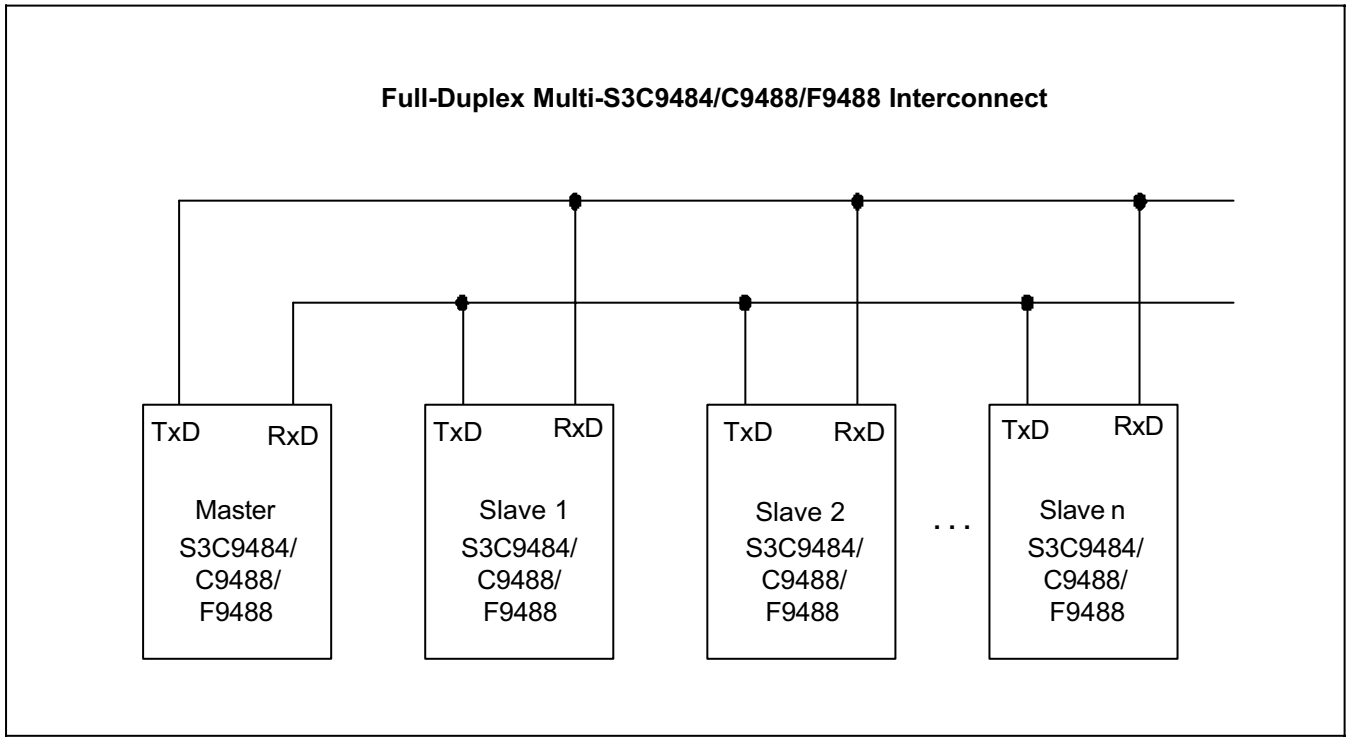


Figure 12-9. Connection Example for Multiprocessor Serial Data Communications

13 WATCH TIMER

OVERVIEW

Watch timer functions include real-time and watch-time measurement and interval timing for the system clock. To start watch timer operation, set bit 1 and bit 6 of the watch timer mode register, WTCON.1 and .6, to "1". After the watch timer starts and elapses a time, the watch timer interrupt is automatically set to "1", and interrupt requests commence in 3.9ms, 0.25 s, 0.5s or 1.0s intervals.

The watch timer can generate a steady 0.5kHz, 1kHz, 2 kHz or 4 kHz signal to the BUZZER output. By setting WTCON.3 and WTCON.2 to "11b", the watch timer will function in high-speed mode, generating an interrupt every 3.91 ms. High-speed mode is useful for timing events for program debugging sequences.
The watch timer supplies the clock frequency for the LCD controller (f_{LCD}). Therefore, **if the watch timer is disabled, the LCD controller does not operate**.

— Real-Time and Watch-Time Measurement
— Using a Main System or Subsystem Clock Source
— Clock Source Generation for LCD Controller
— Buzzer Output Frequency Generator
— Timing Tests in High-Speed Mode

WATCH TIMER CONTROL REGISTER (WTCON)

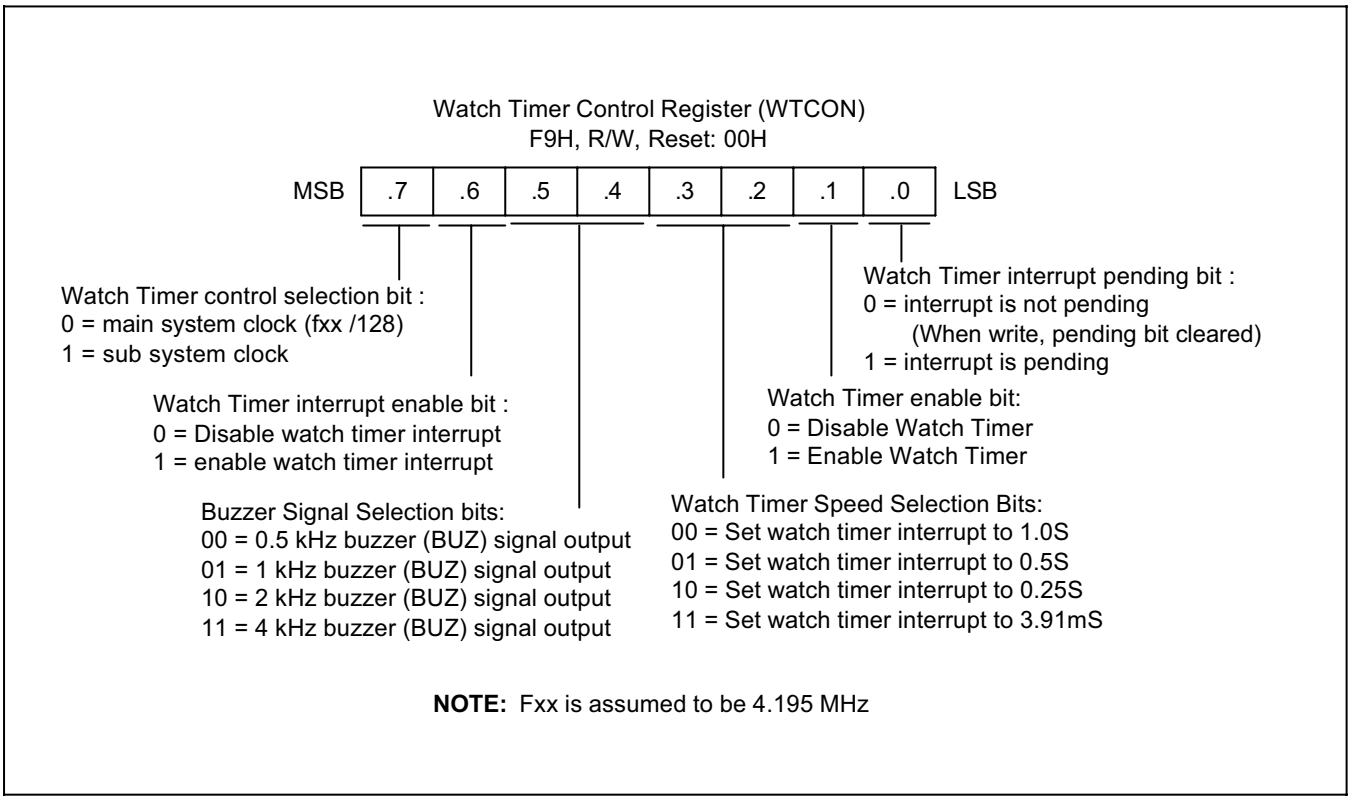

Watch Timer Control Register (WTCON)
F9H, R/W, Reset: 00H
MSB
.7
.6
.5
.4
.3
.2
.1
.0
LSB
Watch Timer control selection bit :
0 = main system clock (fxx /128)
1 = sub system clock
Watch Timer interrupt enable bit :
0 = Disable watch timer interrupt
1 = enable watch timer interrupt
Buzzer Signal Selection bits:
00 = 0.5 kHz buzzer (BUZ) signal output
01 = 1 kHz buzzer (BUZ) signal output
10 = 2 kHz buzzer (BUZ) signal output
11 = 4 kHz buzzer (BUZ) signal output
Watch Timer Speed Selection Bits:
00 = Set watch timer interrupt to 1.0S
01 = Set watch timer interrupt to 0.5S
10 = Set watch timer interrupt to 0.25S
11 = Set watch timer interrupt to 3.91mS
Watch Timer enable bit:
0 = Disable Watch Timer
1 = Enable Watch Timer
Watch Timer interrupt pending bit :
0 = interrupt is not pending
(When write, pending bit cleared)
1 = interrupt is pending
NOTE: Fxx is assumed to be 4.195 MHz

Figure 13-1. Watch Timer Control Register (WTCON)

WATCH TIMER CIRCUIT DIAGRAM

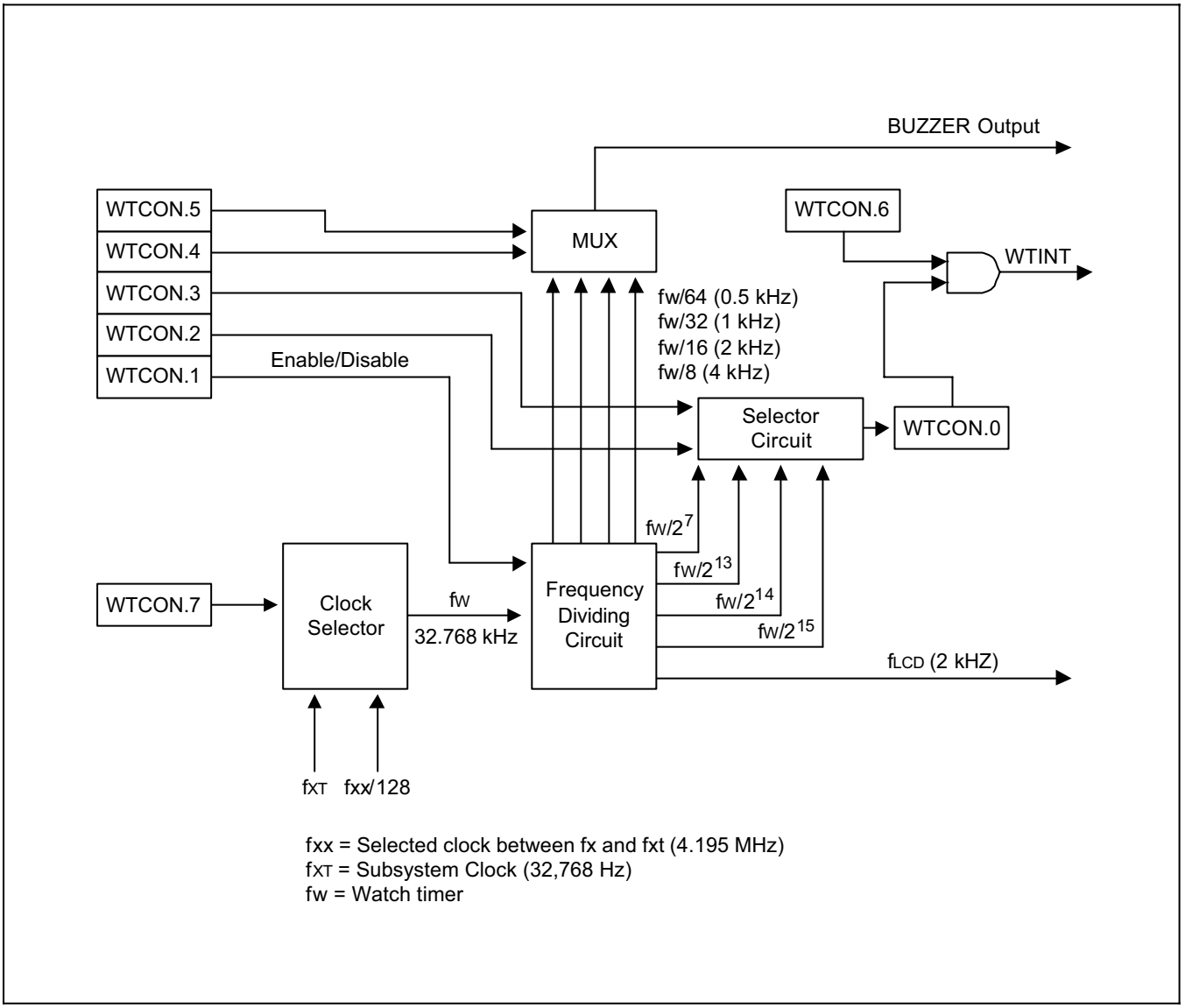


Figure 13-1. Watch Timer Circuit Diagram

+ PROGRAMMING TIP – Using The WATCH TIMER Display (3.91ms,4kHz buzzer out)

```
        .INCLUDE        "C:\SKSTUDIO\INCLUDE\REG\S3C9488.REG"
        VECTOR          00H,F9488_INT

        .ORG            003CH
            DB          0FFH
            DB          0FFH
            DB          01100000B           ;DISABLE LVR
            DB          00000011B           ;SUB OSCILLATOR,BT OVERFLOW, RESET PIN ENALBE

        .ORG            100H

RESET:

            DI
            LD          WDTCON,#10101010B
            LD          BTCON,#0001011B
            LD          CLKCON,#00011000B
            LD          SP,#0C0H
            LD          SYM,#00H
            LD          OSCCON,#00000000B
            LD          P1CONL,#10100110B   ;BUZZER OUTPUT
            LD          WTCON,#11111110B    ;SUB SYSTEM CLOCK, 4KHz,3.91ms interval
            EI

;===============================================================================

MAIN
            JP          MAIN
;===============================================================================

F9488_INT
            TM          WTCON,#01H          ;CHECK WHAT INTERRUPT PENDING BIT IS SET
            JP          NZ,WATCH_T_INT

            ;.....

            IRET

WATCH_T_INT
            AND         WTCON,#0FEH
            XOR         P1,#01H             ;PORT TOGGLE WHENEVER INTERRUPT SERVICE
                                            ;ROUTINE IS EXECUTED
            NOP
            NOP
            IRET
.END
```

14 LCD CONTROLLER / DRIVER

OVERVIEW

The S3C9484/C9488/F9488 micro-controller can directly drive an up-to-19-digit (19-segment) LCD panel. The LCD module has the following components:

— LCD controller/driver

— Display RAM (00H-12H) for storing display data in page 1

— 19 segment output pins (SEG0 – SEG18)

— 8 common output pins (COM0 - COM7)

Bit settings in the LCD control register, LCDCON, determine the LCD frame frequency, duty and bias, and the segment pins used for display output. When a subsystem clock is selected as the LCD clock source, the LCD display is enabled even during stop and idle modes.

The LCD Voltage control register LCDVOL switches contrast output to segment/port.
LCD data stored in the display RAM locations are transferred to the segment signal pins automatically without program control.

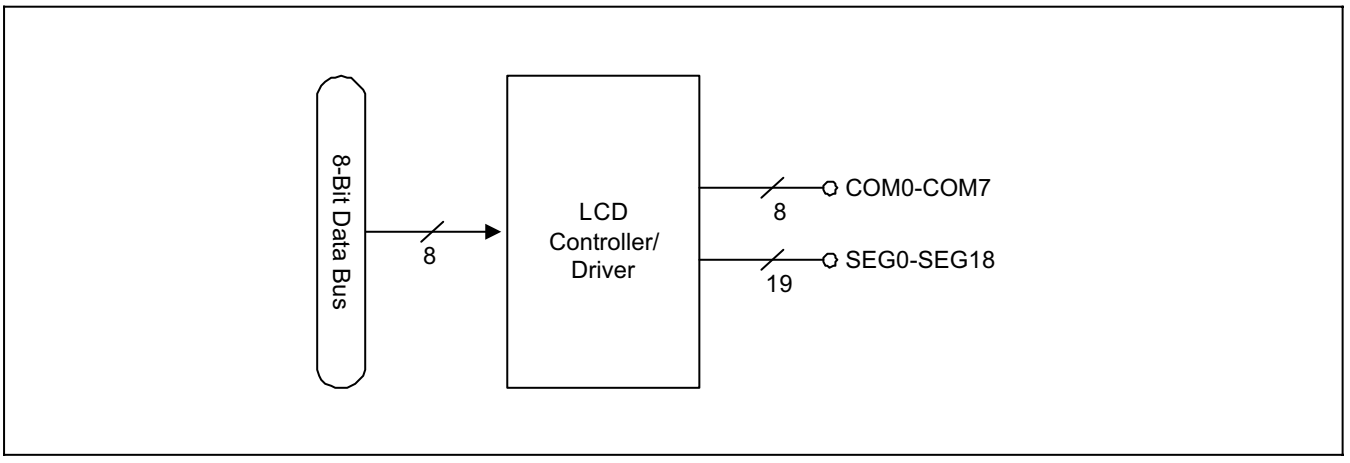


Figure 14-1. LCD Function Diagram

LCD CIRCUIT DIAGRAM

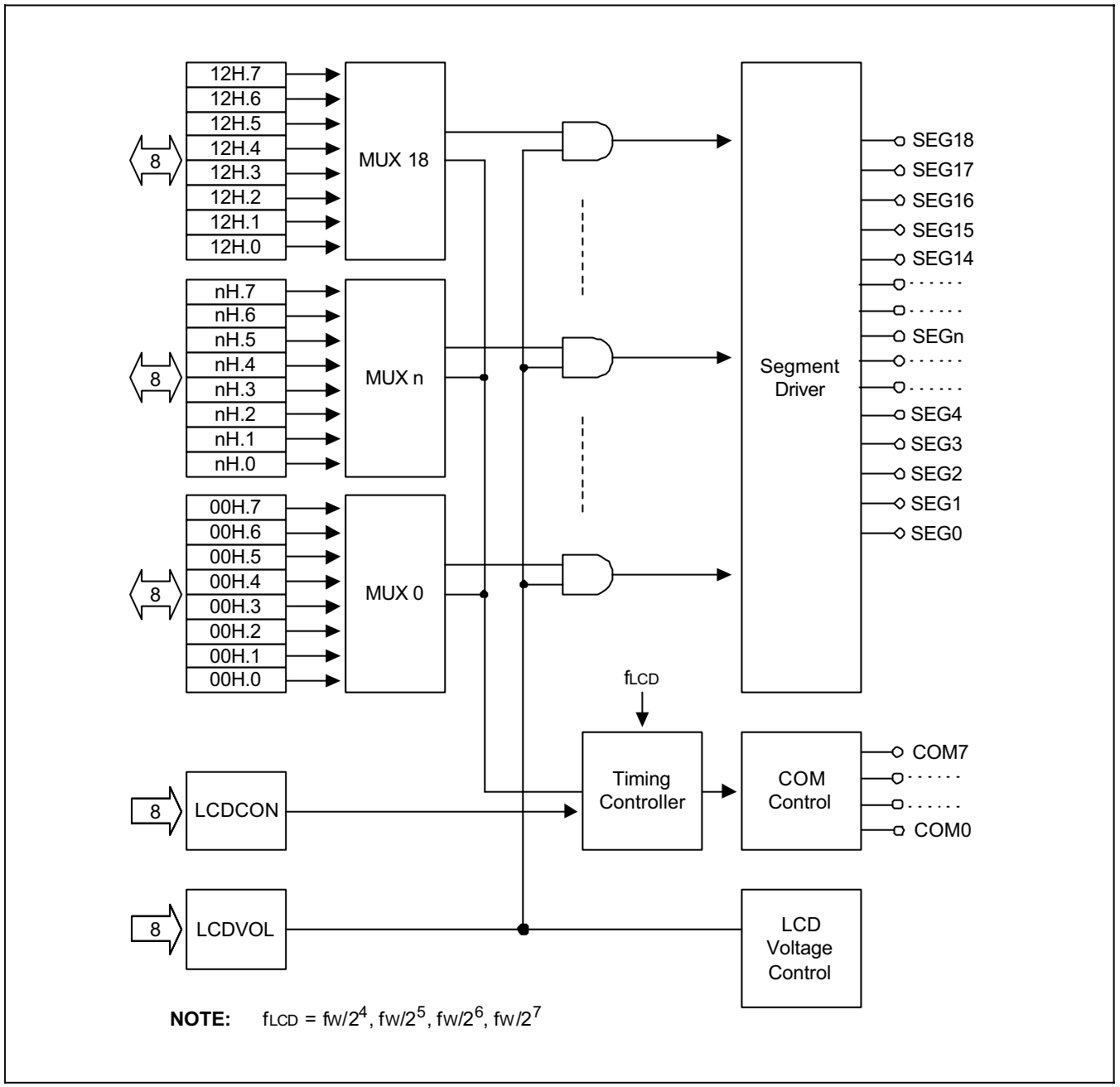


Figure 14-2. LCD Circuit Diagram

LCD RAM ADDRESS AREA

RAM addresses 00H-12H of page 1 are used as LCD data memory. When the bit value of a display segment is "1", the LCD display is turned on; when the bit value is "0", the display is turned off.

Display RAM data are sent out through segment pins SEG0-SEG18 using a direct memory access (DMA) method that is synchronized with the f_{LCD} signal. If these RAM addresses not used for LCD display, you can be allocated to general-purpose use.

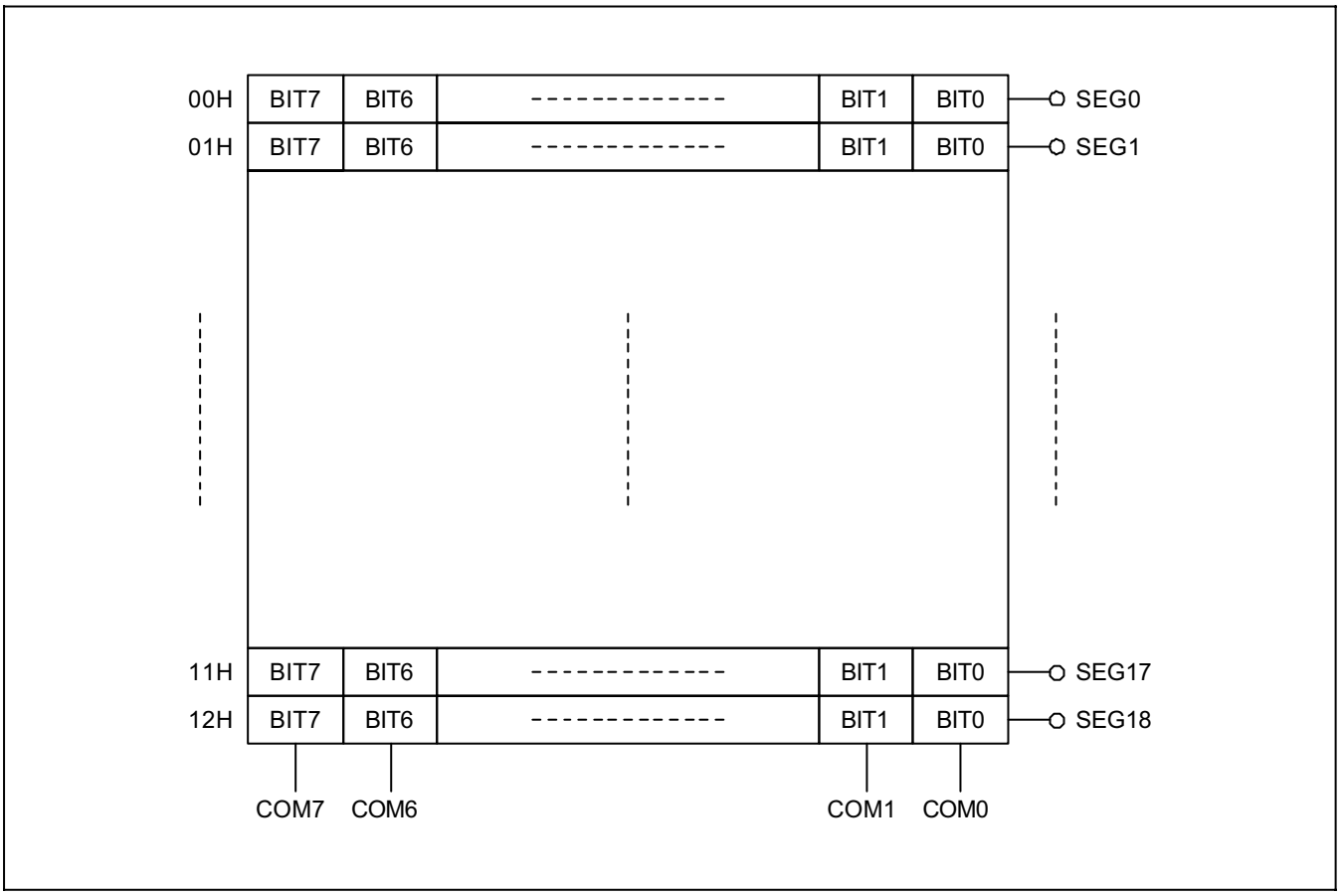


Figure 14-3. LCD Display Data RAM Organization

NOTE

In MDS(such as SK-1000), before changing PAGE(PAGE0 → PAGE1), you must disable global interrupt(DI) and during accessing PAGE1, you don't have to use "CALL" instruction.

LCD CONTROL REGISTER (LCDCON), D0H

The LCD control register LCDCON is mapped to RAM addresses D0H. LCDCON controls these LCD functions:

— LCD module enable/disable control (LCDCON.7)

— LCD Duty and Bias selection (LCDCON.5- LCDCON.4)

— LCD dot on/off control bit (LCDCON.3- LCDCON.2)

— LCD clock frequency selection (LCDCON.1- LCDCON.0)

The LCD clock signal determines the frequency of COM signal scanning of each segment output. This is also referred to as the 'frame frequency' Since LCD clock is generated by dividing the watch timer clock (fw), the watch timer must be enabled when the LCD display is turned on. RESET clears the LCDCON register values to logic zero. This produces the following LCD control settings:

— LCD clock frequency is the watch timer clock (fw)/2^7 = 256 Hz

The LCD display can continue to operate during idle and stop modes if a subsystem clock is used as the watch timer source.

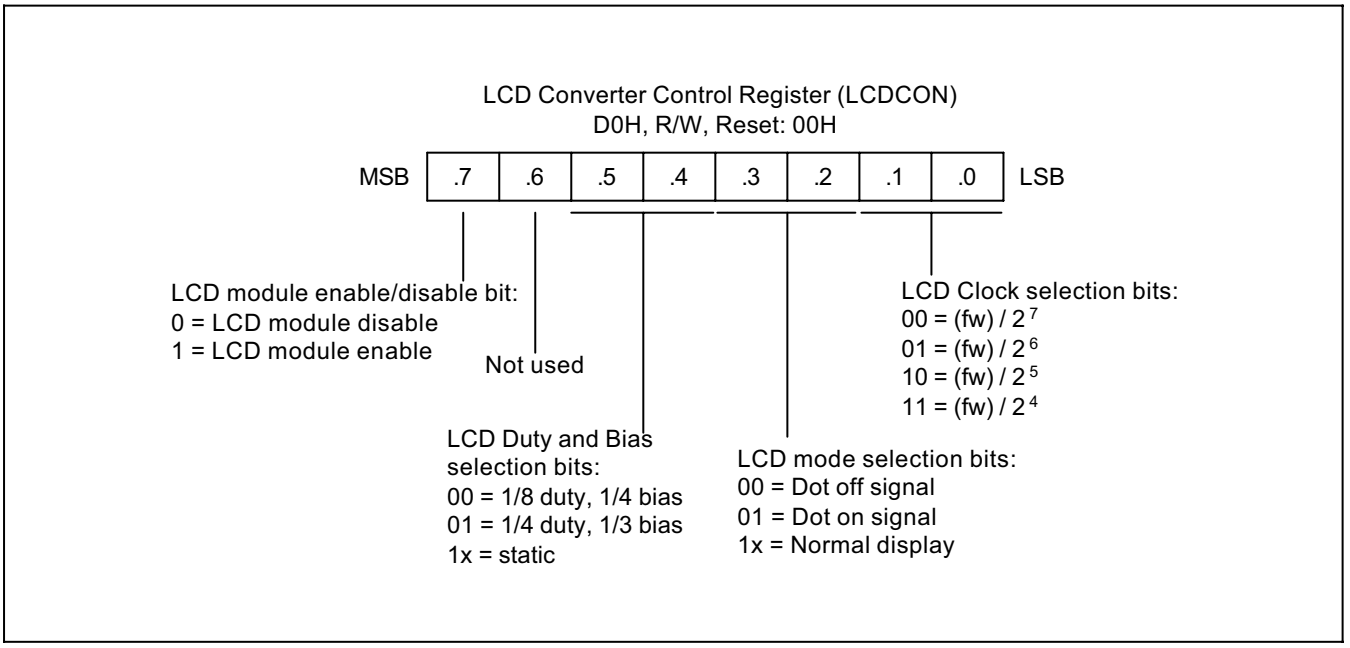


Figure 14-4. LCD Control Register (LCDCON)

LCD VOLTAGE CONTROL REGISTER (LCDVOL)

The LCD Voltage control register LCDVOL is mapped to RAM addresses D1H.
LCDVOL is used to control the LCD contrast up to 16 step contrast level.

— LCD contrast control enable/disable bit (LCDVOL.7)

— LCD contrast segment output selection bits (LCDVOL.0 -LCDVOL.3)

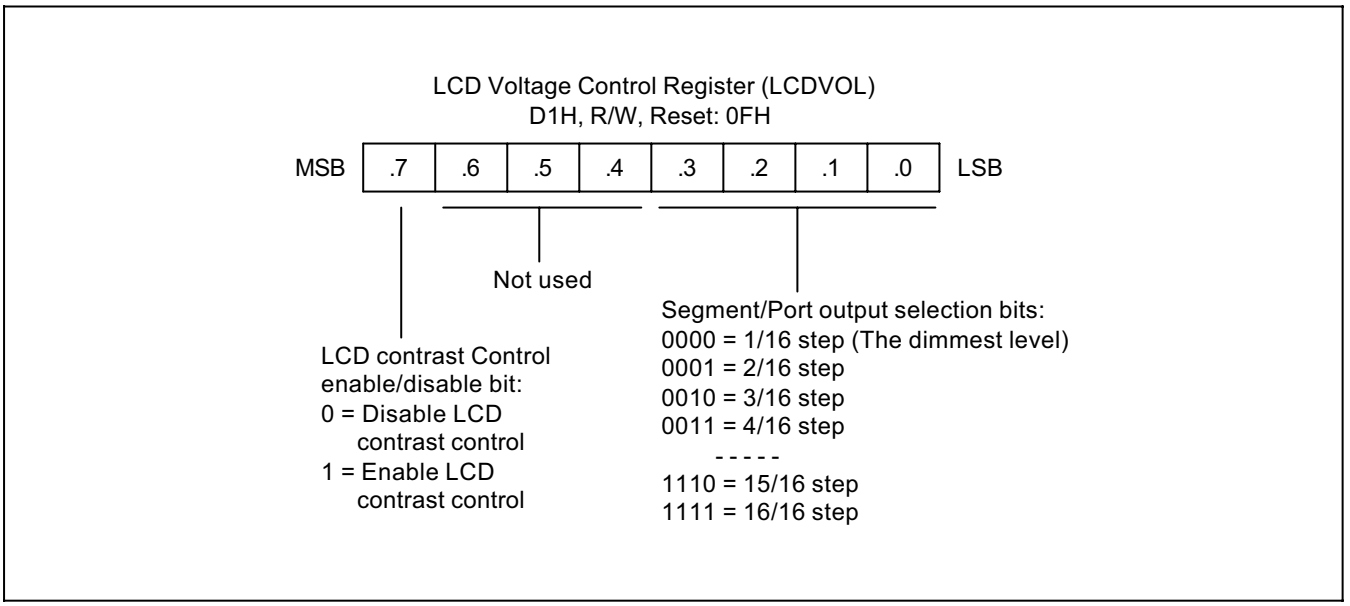
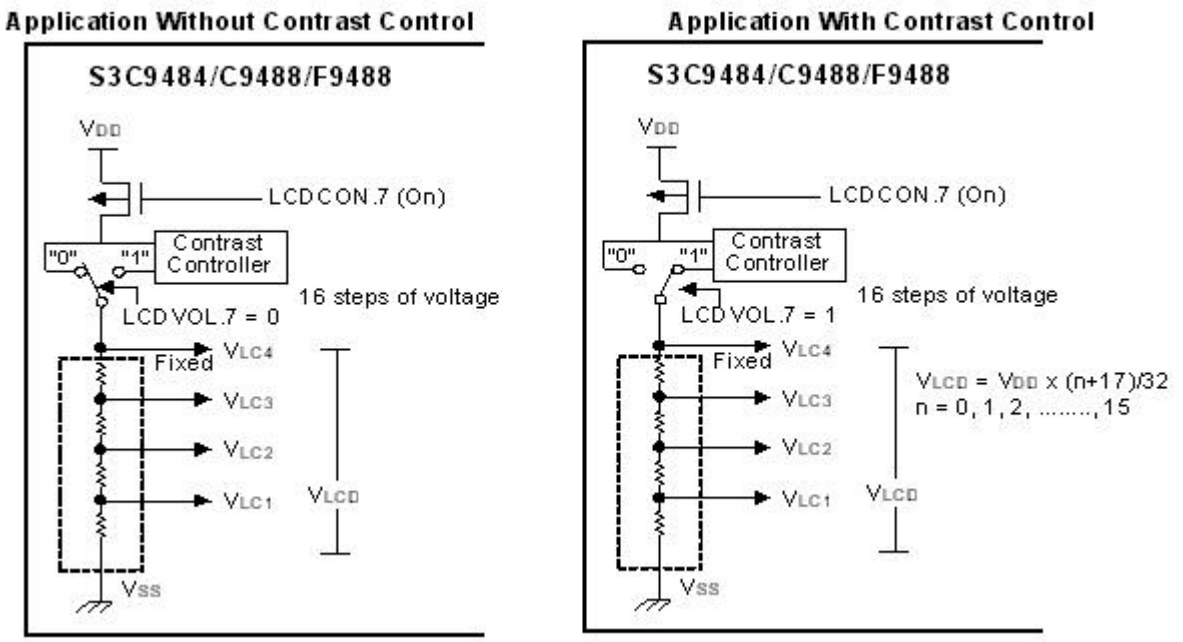
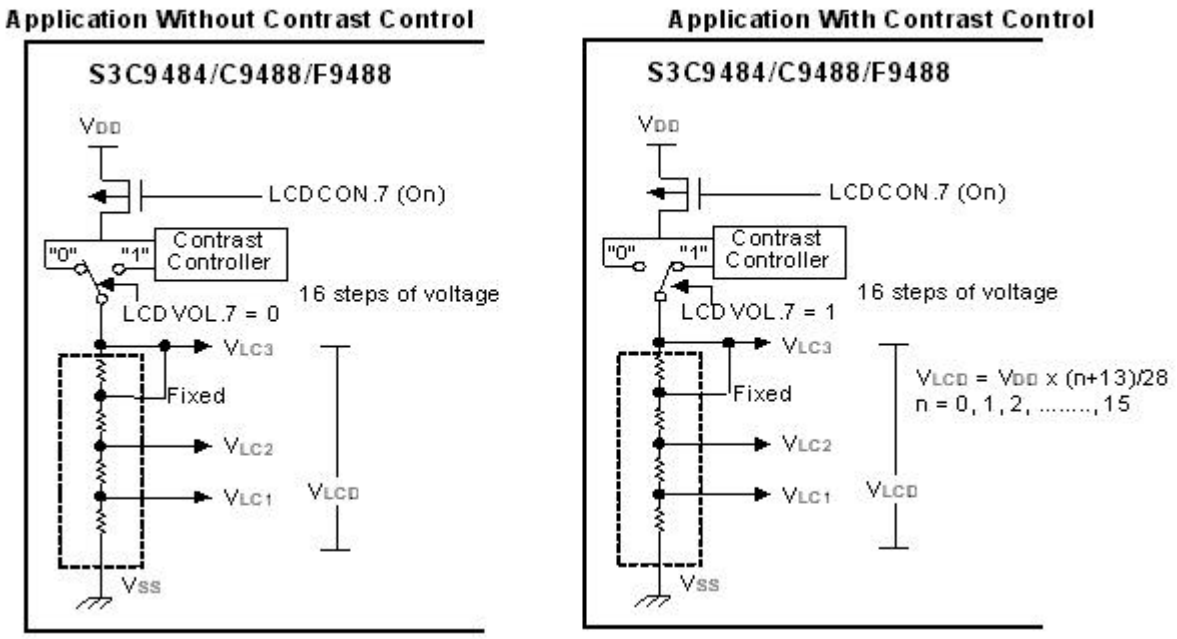


Figure 14-5. LCD Drive Voltage Control Register (LCDVOL)



NOTE: When LCDVOL.7 is logic one, you can control LCD contrast by writing data to LCDVOL.3-.0

Figure 14-6. Internal Voltage Dividing Resistor Connection (1/4 Bias, Display On)



NOTE: When LCDVOL.7 is logic one, you can control LCD contrast by writing data to LCDVOL.3-.0

Figure 14-7. Internal Voltage Dividing Resistor Connection (1/3 Bias, Display On)

LCD DRIVE VOLTAGE

The LCD display is turned on only when the voltage difference between the common and segment signals is greater than V_{LCD}. The LCD display is turned off when the difference between the common and segment signal voltages is less than $V_{LCD.}$ The turn-on voltage, + V_{LCD} or - V_{LCD}, is generated only when both signals are the selected signals of the bias. Table 14-1 shows LCD drive voltages level for static mode, 1/3 bias, 1/4 bias.

Table 14-1. LCD Drive Bias Voltages Level Values

LCD Power Supply	Static Mode	1/3 Bias	1/4 Bias
V_{LC4}	V_{LCD}	–	V_{LCD}
V_{LC3}	–	V_{LCD}	3/4 V_{LCD}
V_{LC2}	–	2/3 V_{LCD}	2/4 V_{LCD}
V_{LC1}	–	1/3 V_{LCD}	1/4 V_{LCD}
V_{SS}	0 V	0 V	0 V

NOTE: The LCD panel display may be deteriorated if a DC voltage is applied that lies between the common and segment signal voltage. Therefore, always drive the LCD panel with AC voltage.

LCD SEG/COM SIGNALS

The 19 LCD segment signal pins are connected to corresponding display RAM locations at 00H-12H. Bits 0-7 of the display RAM are synchronized with the common signal output pins COM0, ... , and COM7.

When the bit value of a display RAM location is "1", a select signal is sent to the corresponding segment pin. When the display bit is "0", a 'no-select' signal is sent to the corresponding segment pin. Each bias has select and no-select signals.

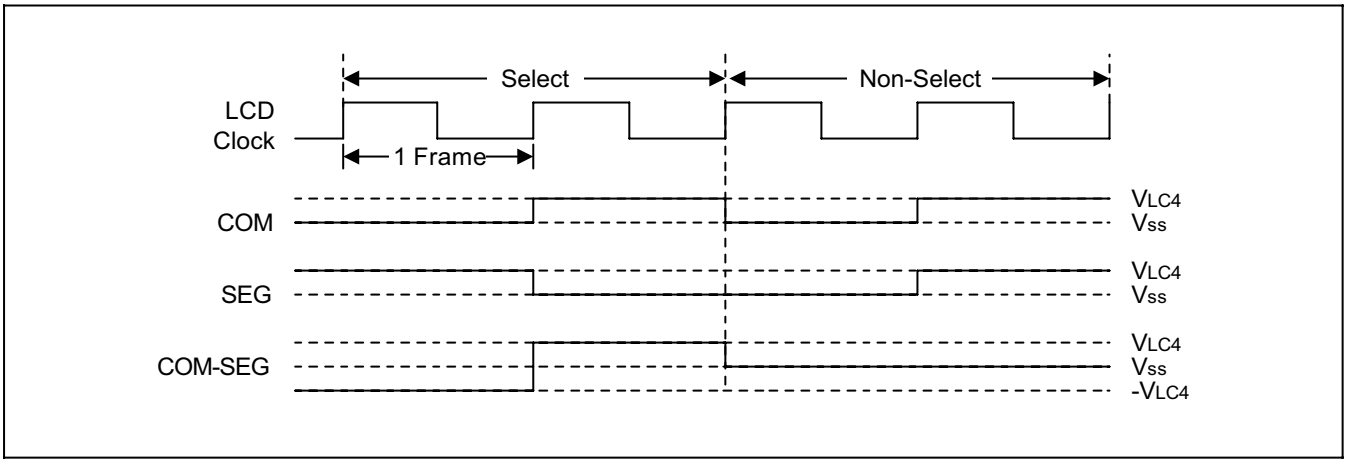


Figure 14-8. Select/No-Select Bias Signals in Static Display Mode

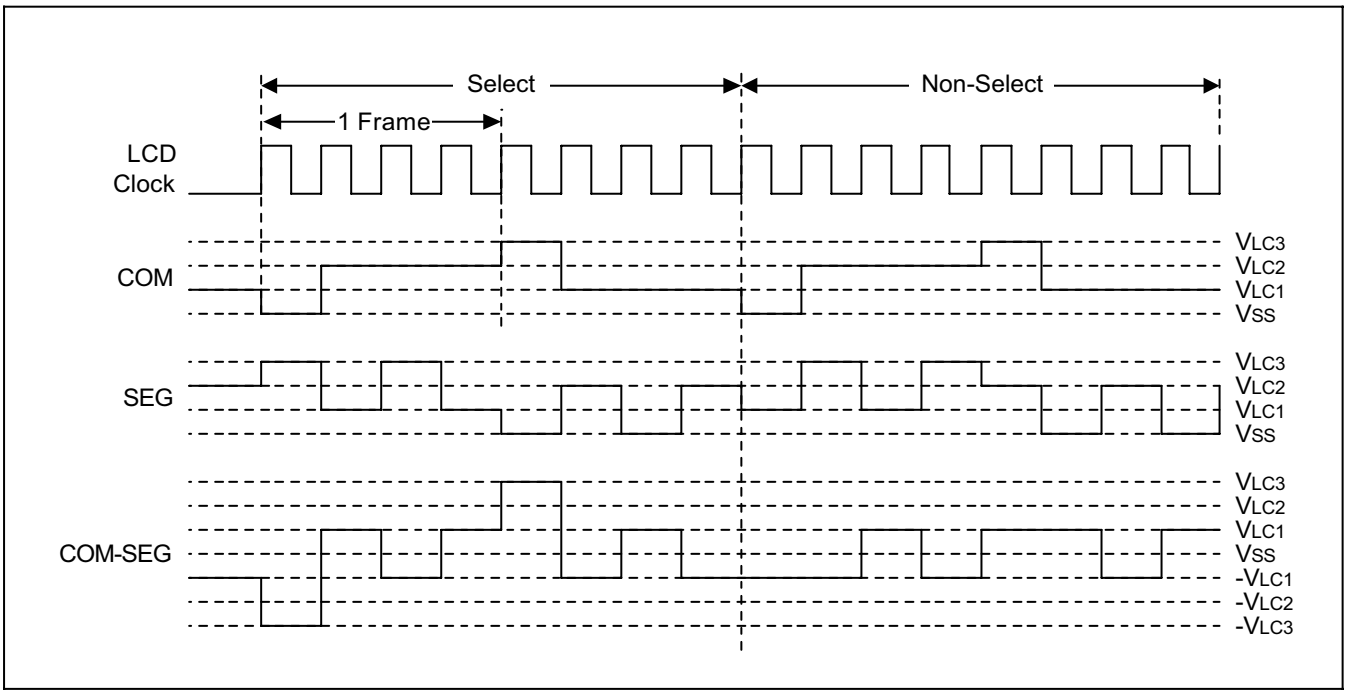


Figure 14-9. Select/No-Select Bias Signals in 1/4 Duty, 1/3 Bias Display Mode

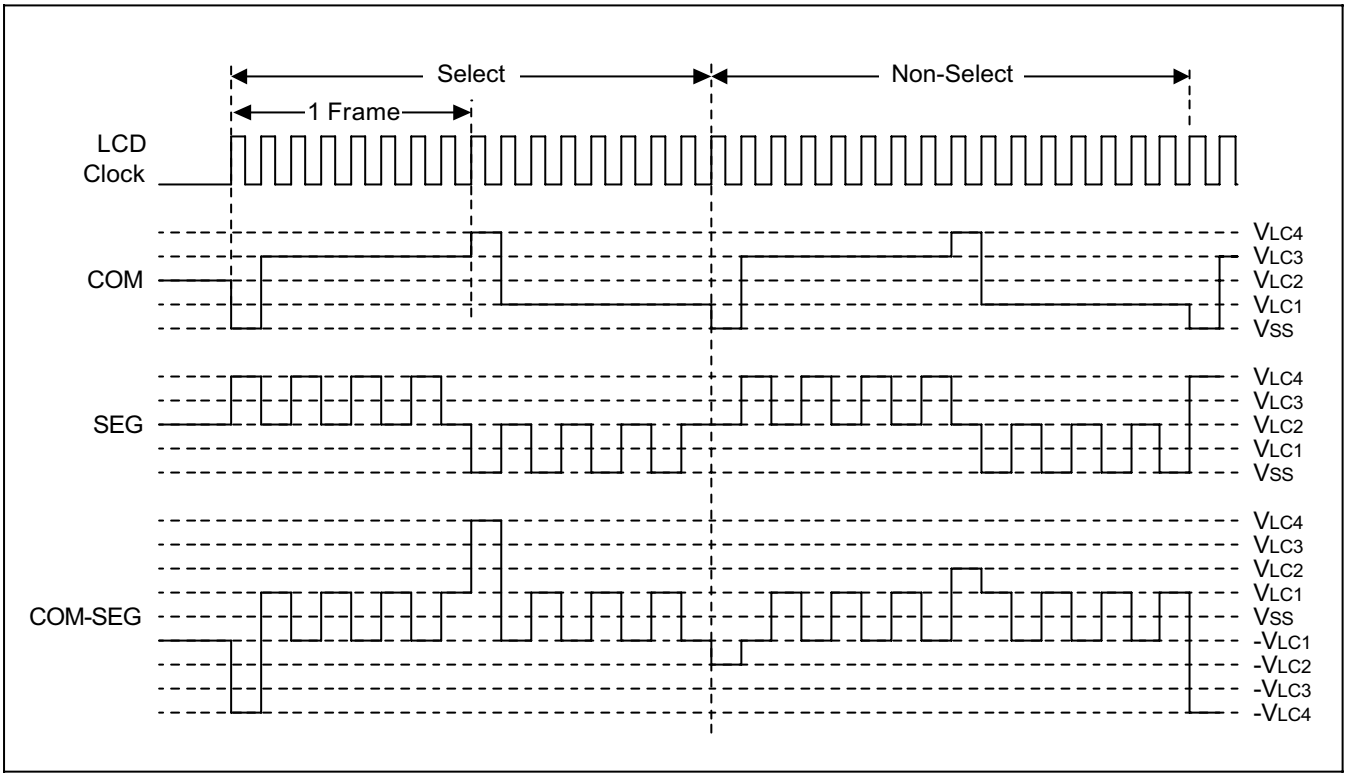


Figure 14-10. Select/No-Select Bias Signals in 1/8 Duty, 1/4 Bias Display Mode

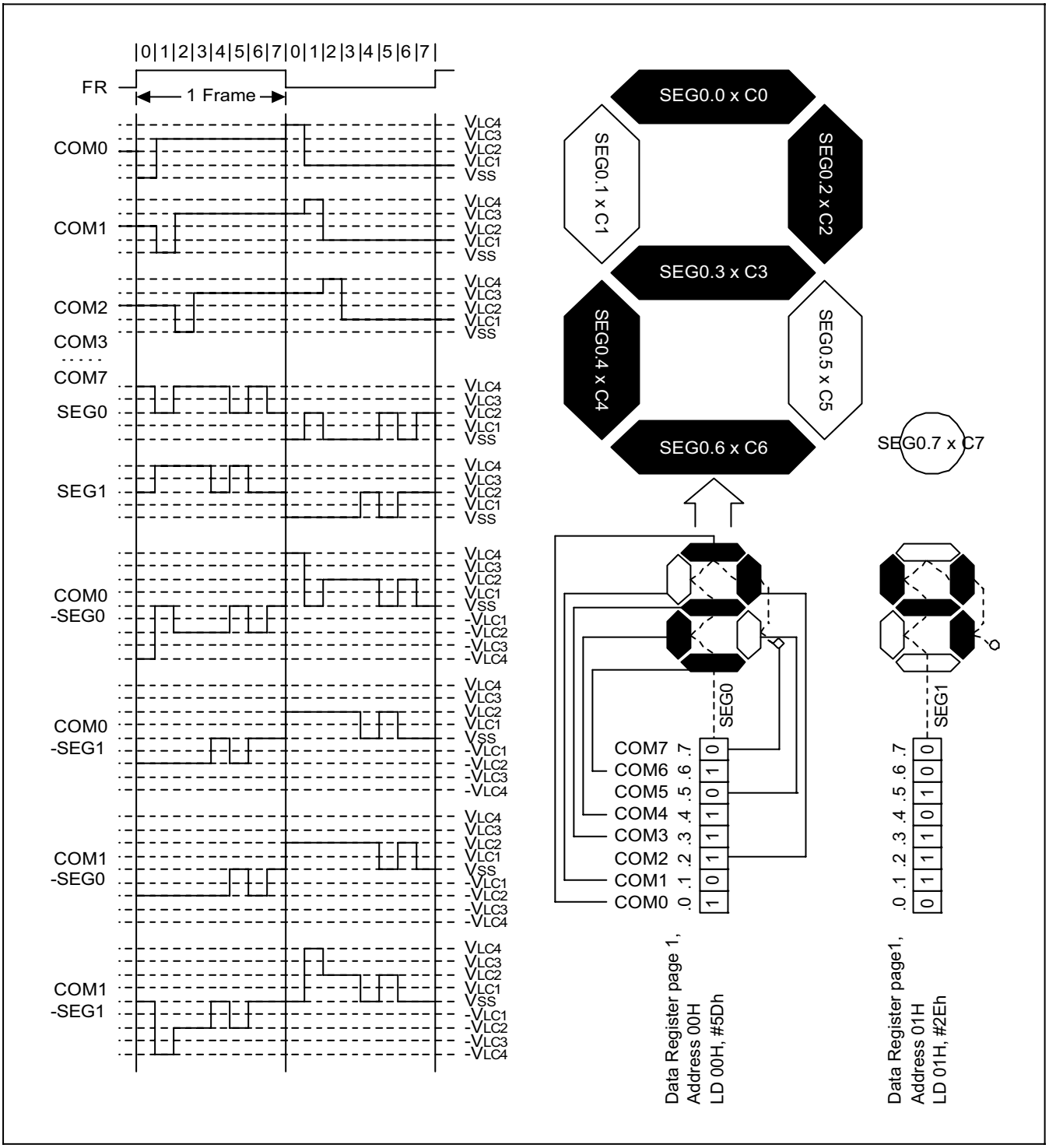


Figure 14-11. LCD Signal and Wave Forms Example in 1/8 Duty, 1/4 Bias Display Mode

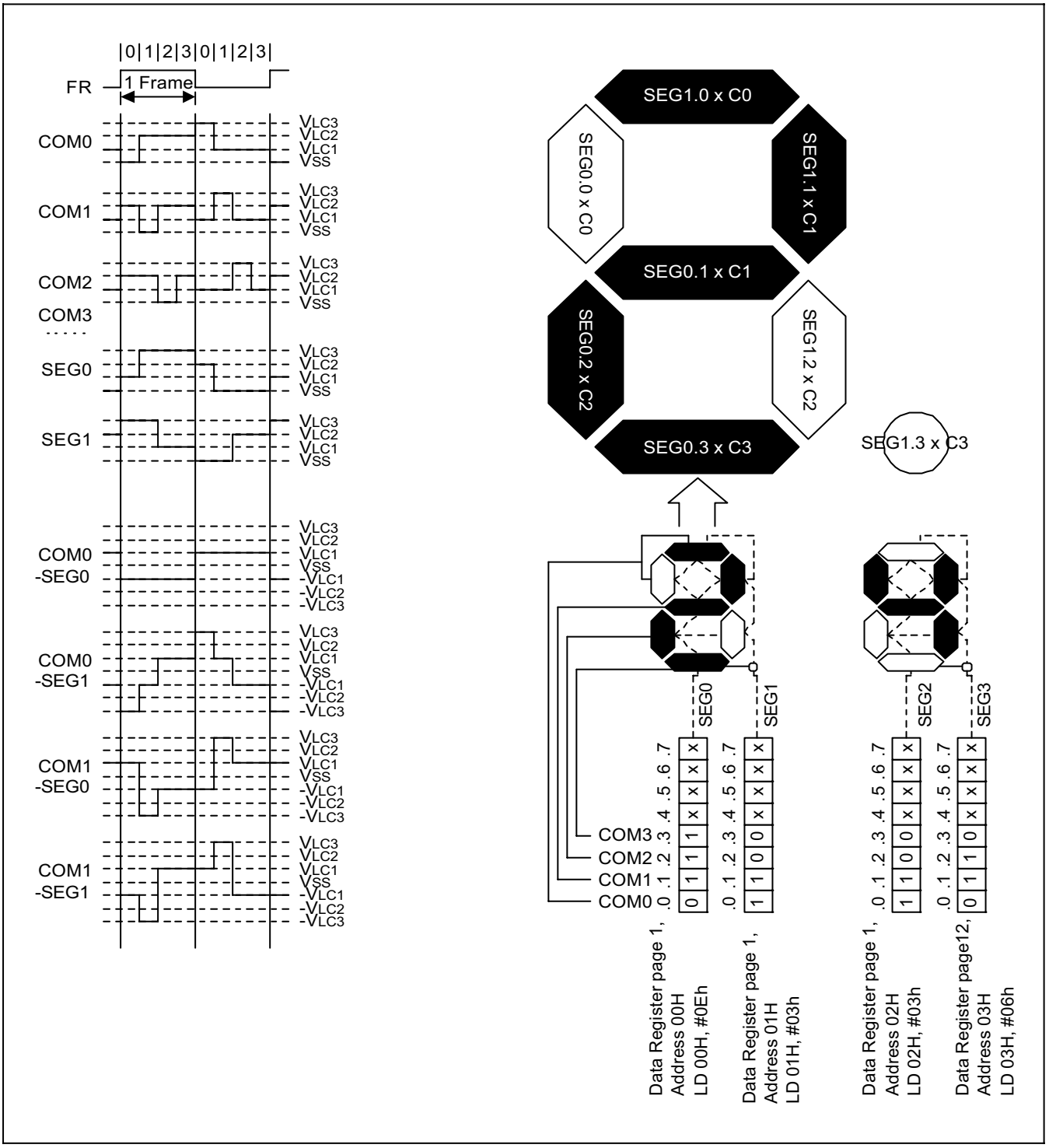


Figure 14-12. LCD Signals and Wave Forms Example in 1/4 Duty, 1/3 Bias Display Mode

+ PROGRAMMING TIP – Using The LCD Display

```
            .INCLUDE    "C:\SKSTUDIO\INCLUDE\REG\S3C9488.REG"

LCD_DATA0_P1            .EQU    00H

            .ORG        003CH
                        DB      0FFH
                        DB      0FFH
                        DB      01100000B
                        DB      00000011B                   ;Smart Option setting

            .ORG        100H

RESET:

            DI
            LD          WDTCON,#10101010B
            LD          BTCON,#0001011B
            LD          CLKCON,#00011000B
            LD          SP,#0C0H
            LD          SYM,#00H
            LD          OSCCON,#00000000B
            LD          LCDCON,#10001000B   ;1/8 duty,1/4 bias,fw/128
            LD          LCDVOL,#10001111B   ;lcd contrast enable,16/16 step
            LD          P0CONH,#0FFH        ;COM4-COM7
            LD          P0CONL,#11101010B
            LD          P1CONH,#0FFH        ;COM0-COM3
            LD          P1PUR,#00H
            LD          P2CONH,#0FFH        ;SEG7-SEG10
            LD          P2CONL,#0FFH        ;SEG3-SEG6
            LD          P3CONH,#10101011B   ;SEG18
            LD          P3CONL,#11111111B   ;SEG15-SEG17
            LD          P4CONH,#00111111B   ;SEG12-SEG14
            LD          P4CONL,#0FFH        ;SEG0-SEG2,SEG11
            LD          WTCON,#02H          ;Watch Timer enable
;=============================================================================
```

```
MAIN

            LD      SYM,#01H                    ;SELECT PAGE1
            LD      R0,#LCD_DATA0_P1    ;LOAD LCD DISPLAY DATA RAM0
            LD      R2,#0
            LD      R3,#0

LOOP        LDC     R1,#LCD_DATA[RR2]
            LD      @R0,R1
            INC     R0
            INC     R3
            CP      R3,#13H
            JP      C,LOOP
            LD      SYM,#00H                    ;SELECT PAGE0
            JP      $

LCD_DATA
            .DB     00H,48H,34H,0D0H,22H,11H,89H,0E2H,35H,0FFH
            .DB     77H,33H,67H,99H,46H,0F1H,4H,88H,54H

;=============================================================================
.END
```


SAMSUNG ELECTRONICS

15 10-BIT ANALOG-TO-DIGITAL CONVERTER

OVERVIEW

The 10-bit A/D converter (ADC) module uses successive approximation logic to convert analog levels entering at one of the nine input channels to equivalent 10-bit digital values. The analog input level must lie between the AV_{REF} and V_{SS} values. The A/D converter has the following components:

— Analog comparator with successive approximation logic

— D/A converter logic (resistor string type)

— ADC control register (ADCON)

— Nine multiplexed analog data input pins (AD0 – AD8) , alternately digital data I/O port

— 10-bit A/D conversion data output register (ADDATAH/L)

— AV_{REF} pins, AV_{SS} is internally connected to V_{SS}

FUNCTION DESCRIPTION

To initiate an analog-to-digital conversion procedure, at the first you must set port control register(P0CONH/ P0CONL/P1CONL) for AD analog input. And you write the channel selection data in the A/D converter control register ADCON.4-.6 to select one of the eight analog input pins (AD0-8) and set the conversion start bit, ADCON.0. The read-write ADCON register is located at address FCH. The unused pin can be used for normal I/O.

During a normal conversion, ADC logic initially set the successive approximation register to 200H (the approximate half-way point of an 10-bit register). This register is then updated automatically during each conversion step. The successive approximation block performs 10-bit conversions for one input channel at a time. You can dynamically select different channels by manipulating the channel selection bit value (ADCON.7 - 4) in the ADCON register. To start the A/D conversion, you should set the enable bit, ADCON.0. When a conversion is completed, the end-of-conversion (EOC) bit is automatically set to 1 and the result is dumped into the ADDATAH/L register where it can be read. The A/D converter then enters an idle state. Remember to read the contents of ADDATAH/L before another conversion starts. Otherwise, the previous result will be overwritten by the next conversion result.

NOTE

Because the A/D converter **does not use** sample-and-hold circuitry, it is very important that fluctuation in the analog level at the AD0-AD8 input pins during a conversion procedure be kept to an absolute minimum. Any change in the input level, perhaps due to noise, will invalidate the result. **If the chip enters to STOP or IDLE mode in conversion process, there will be a leakage current path in A/D block**. You must use STOP or IDLE mode after ADC operation is finished.

CONVERSION TIMING

The A/D conversion process requires 4 steps (4 clock edges) to convert each bit and 10 clocks to set-up A/D conversion. Therefore, total of 50 clocks are required to complete an 10-bit conversion: When Fxx/8 is selected for conversion clock with an 8 MHz fxx clock frequency, one clock cycle is 1 us. Each bit conversion requires 4 clocks, the conversion rate is calculated as follows:

4 clocks/bit × 10 bits + set-up time = 50 clocks, 50 clock × 1us = 50 us at 1 MHz

A/D CONVERTER CONTROL REGISTER (ADCON)

The A/D converter control register, ADCON, is located at address FCH. It has three functions:

— Analog input pin selection (bits 4, 5, 6, and 7)

— A/D conversion End-of-conversion (EOC) status (bit 3)

— A/D conversion speed selection (bits 1,2)

— A/D operation start (bit 0)

After a reset, the start bit is turned off. You can select only one analog input channel at a time. Other analog input pins (ADC0–ADC8) can be selected dynamically by manipulating the ADCON.4–6 bits. And the pins not used for analog input can be used for normal I/O function.

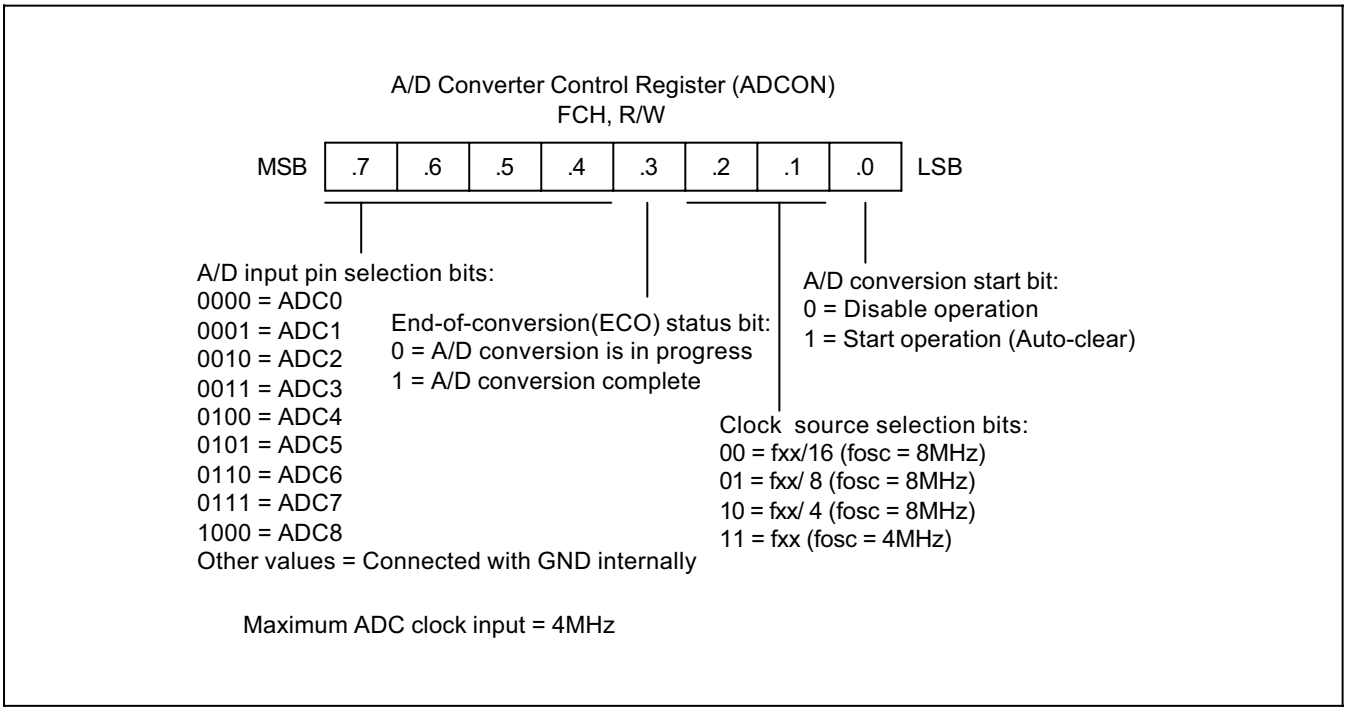


Figure 15-1. A/D Converter Control Register (ADCON)

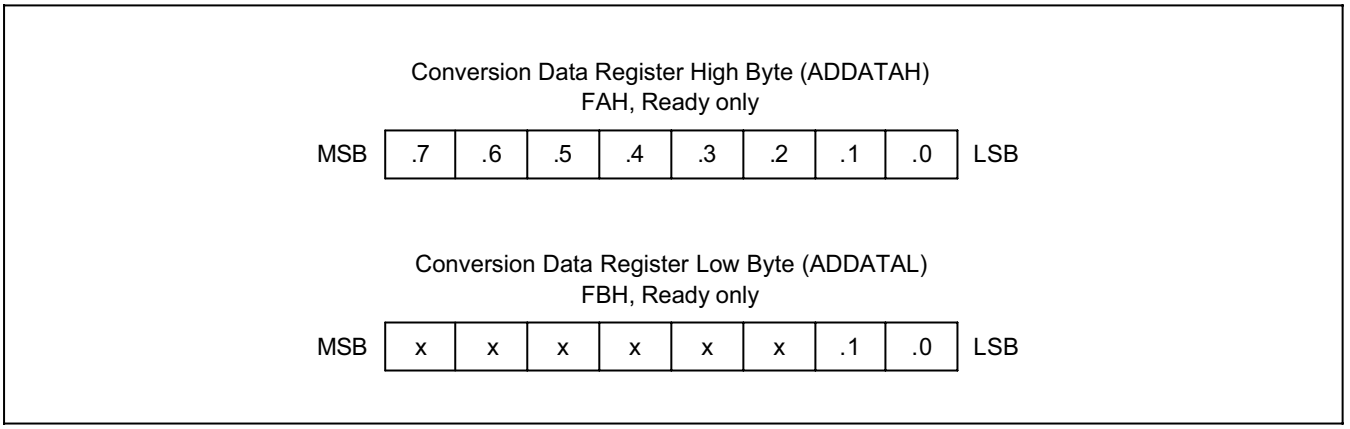


Figure 15-2. A/D Converter Data Register (ADDATAH/L)

INTERNAL REFERENCE VOLTAGE LEVELS

In the ADC function block, the analog input voltage level is compared to the reference voltage. The analog input level must remain within the range V_{SS} to AV_{REF} (usually, AV_{REF} = V_{DD}).

Different reference voltage levels are generated internally along the resistor tree during the analog conversion process for each conversion step. The reference voltage level for the first conversion bit is always 1/2 AV_{REF}.

BLOCK DIAGRAM

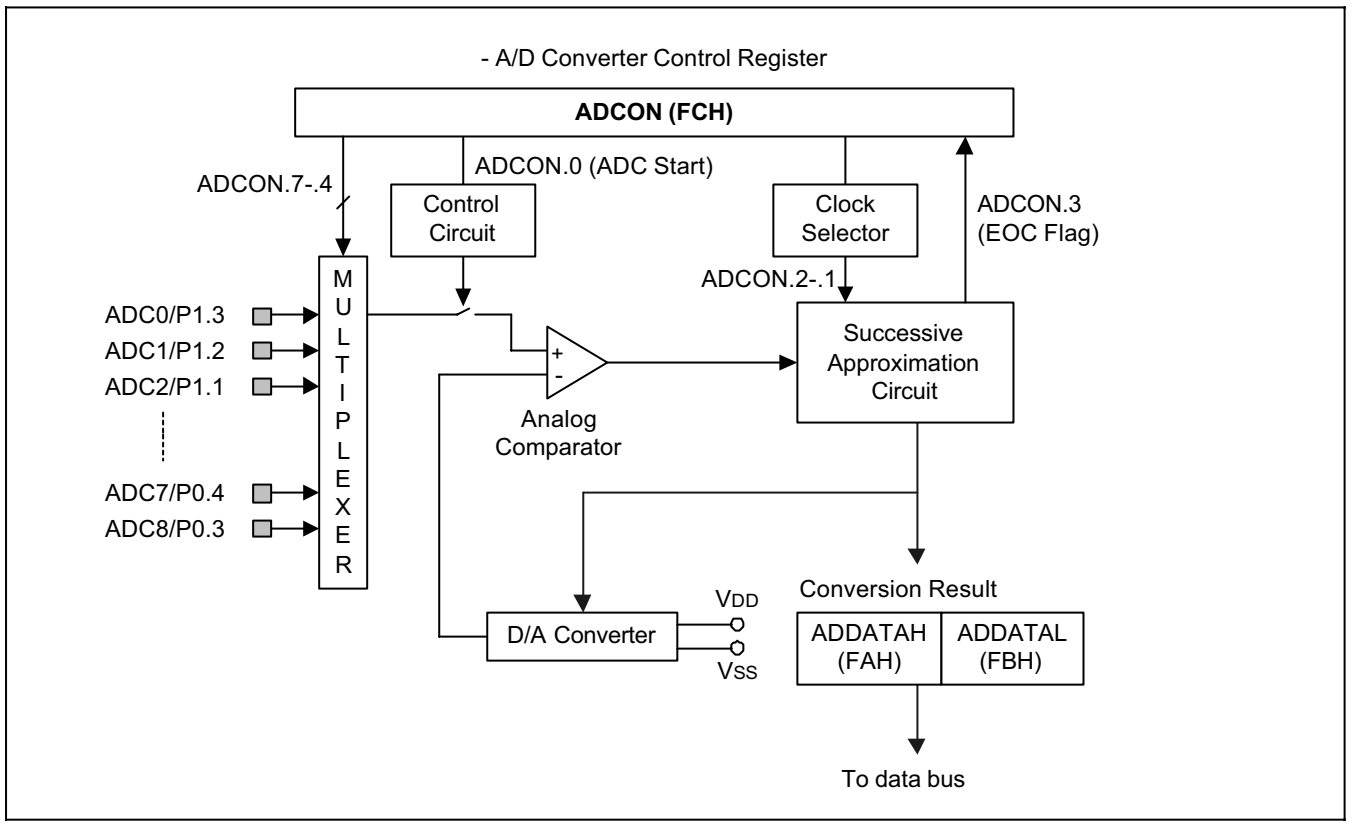
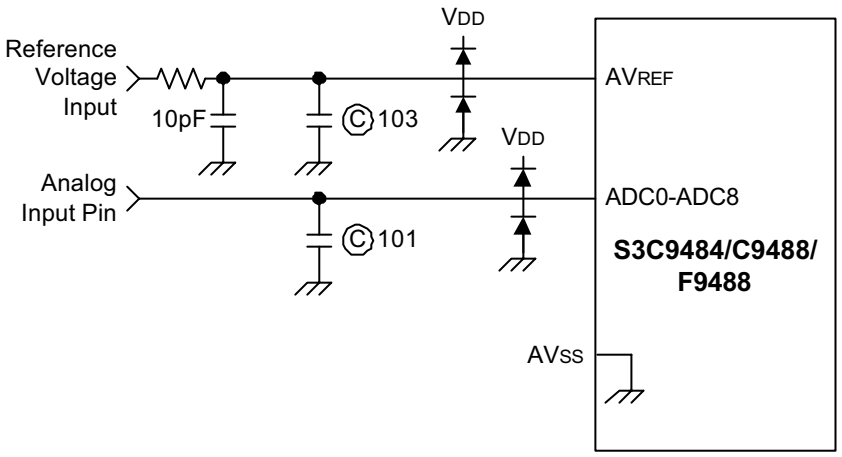


Figure 15-3. A/D Converter Functional Block Diagram

INTERNAL A/D CONVERSION PROCEDURE

1. Analog input must remain between the voltage range of VSS and AVREF.
2. Configure P0.3–P0.7 and P1.0–P1.3 for analog input before A/D conversions. To do this, you have to load the appropriate value to the P0CONH, P0CONL and P1CONL (for ADC0–ADC8) registers.
3. Before the conversion operation starts, you must first select one of the eight input pins (ADC0–ADC8) by writing the appropriate value to the ADCON register.
4. When conversion has been completed, (50 clocks have elapsed), the EOC, ADCON.3 flag is set to "1", so that a check can be made to verify that the conversion was successful.
5. The converted digital value is loaded to the output register, ADDATAH (8-bit) and ADDATAL (2-bit), then the ADC module enters an idle state.
6. The digital conversion result can now be read from the ADDATAH and ADDATAL register.



NOTE: The symbol 'R' signifies an offset resistor with a value of from 50 to 100.
If this resistor is omitted, the absolute accuracy will be maximum of 3 LSBs.

Figure 15-4 Recommended A/D Converter Circuit for Highest Absolute Accuracy

NOTES

16 WATCHDOG TIMER

OVERVIEW

WHATCHDOG TIMER

You can use the watchdog timer :

— Watchdog timer provides an automatic reset mechanism with counter clock source of internal RC ring oscillation or basic timer overflow signal.

— Watchdog timer can run in unintentional STOP/IDLE mode with internal RC ring oscillator. This prevents MCU from remaining in the abnormal STOP/IDLE mode.

The functional components of the watchdog timer block are:

— Internal RC oscillation or basic timer overflow signal.

— Smart Option 3FH.1 selects counter clock source, 16bit watchdog timer overflow condition (bit15 OVF with internal ring oscillator or bit3 OVF with basic timer overflow). Also, on STOP and IDLE mode with internal RC ring oscillator, watchdog timer counter is not cleared by smart option.

— Watchdog timer control register, WDTCON (E5H, read/write)

— 16bit Watchdog Timer Counter

WATCHDOG TIMER CONTROL REGISTER (WDTCON)

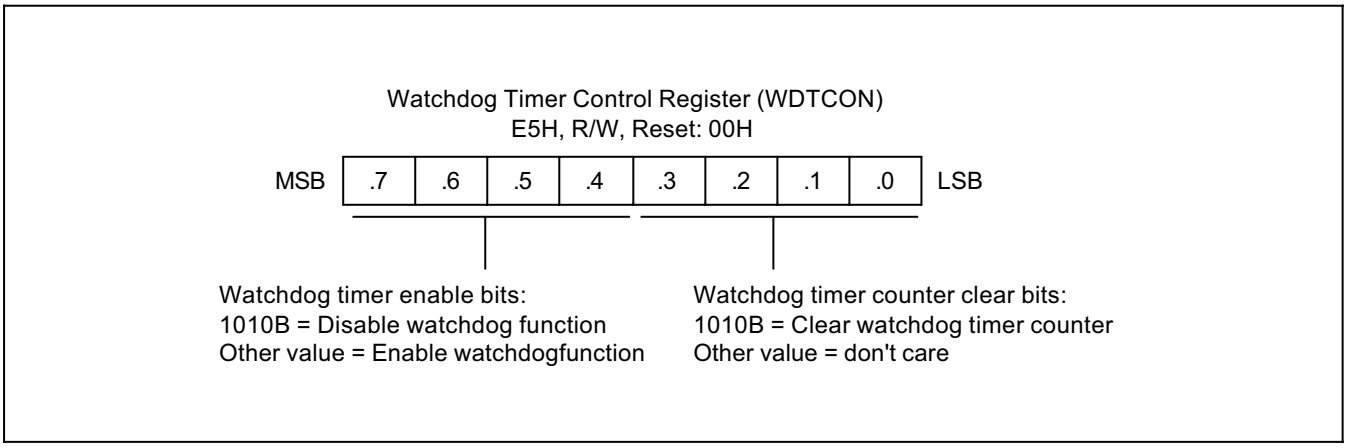


Figure 16-1. Watchdog Timer Control Register (WDTCON)

WATCHDOG TIMER FUNCTION DESCRIPTION

Watchdog Timer Function

You can program the watchdog timer overflow signal (WDTOVF) to generate a reset by setting WDTCON.7-.4 to any value other than "1010B". (The "1010B" value disables the watchdog function.) A reset clears WDTCON to "00H", automatically enabling the watchdog timer function.

The MCU is reset whenever a watchdog timer counter overflow occurs, During normal operation, the application program must prevent from the overflow, To do this, the WDTCNT value must be cleared (by writing a “1010¡ ± to WDTCON.0-.3) at regular intervals.

If a malfunction occurs due to noise or some other error conditions, the watchdog counter clear operation will not be executed by chip malfunction. So, before long, a watchdog timer overflow reset will occur. After this reset, chip will carry out normal operation again. In other words, during the normal operation, the watchdog timer overflow (bit 3 overflow or bit 15 overflow of the 16-bit watchdog timer counter, WDTCNT) does not occur by a 16bit Watchdog timer counter clear operation.

Watchdog Timer Counter Clock Sources Selection

You can select counter clock source between basic timer overflow signal and internal RC ring oscillator.
If you use basic timer overflow clock source, WDT overflow will occur at the time when counter bit 3 is set. If you use internal RC ring oscillator clock source, WDT overflow will occur at the time when counter bit 15 is set.

Watchdog Timer in STOP/IDLE mode

1. If the basic timer overflow signal is selected for the WDT counter clock source, WDT will be disabled automatically by hardware. So system reset can not occur by WDT. WDT counter is cleared automatically in STOP/IDLE mode. In this case, current consumption is very small.
2. If internal RC ring oscillator is selected for the WDT counter clock source, WDT can be enabled in unintentional STOP/IDLE mode. So system reset can occur by WDT. WDT counter is not cleared in STOP/IDLE mode. So, when abnormal STOP or IDLE mode occurs by noise, MCU can be returned to normal operation by WDT overflow reset. But, at this case, STOP/IDLE mode current consumption becomes larger. If noise problem (like chip entering to unintentional STOP/IDLE mode) is more important, you had better use internal RC ring oscillator.

Before running system, you must select Smart Option (3FH.1) for WDT counter source.
If you select internal RC oscillator, normally, you must set Watchdog Timer to be disable before entering to STOP mode. Because, If WDT is not disabled, reset operation will occur by WDT counter overflow.
If you want to use WDT in STOP/IDLE mode for noise problem, current may drain too much by internal RC oscillation. So, if noise issue is not important, you had better select basic timer overflow signal for WDT counter clock source.

Watchdog Timer Counter Overflow Time for Reset

1. If the basic timer overflow signal is selected for the WDT counter clock source and main clock, Fxx, is 8MHz,

Basic Timer Clock	Fxx/128	Fxx/1024	Fxx/4096
Time for WDT overflow	32.76msec	262msec	1.05sec

2. If internal RC ring oscillator is selected for the WDT counter clock source,

Timer for WDT overflow = (1/3.47)μ sec X 2^{16} = 18.89msec

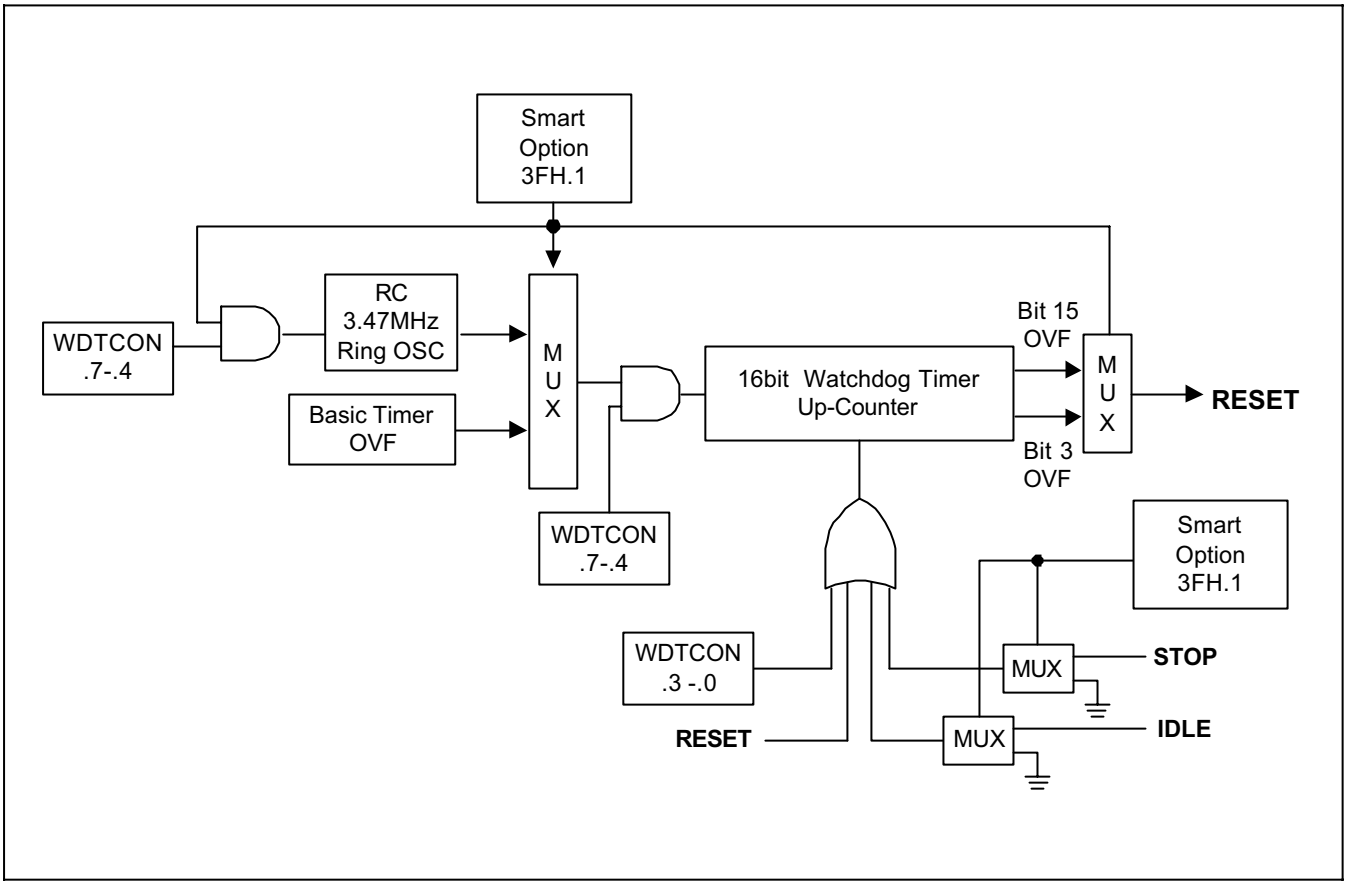


Figure 16-2. Watchdog Timer Block Diagram

NOTES

17 VOLTAGE LEVEL DETECTOR

OVERVIEW

The S3C9484/C9488/F9488 micro-controller has a built-in VLD(Voltage Level Detector) circuit which allows detection of power voltage drop through software. Turning the VLD operation on and off can be controlled by software. Because the IC consumes a large amount of current during VLD operation. It is recommended that the VLD operation should be kept OFF unless it is necessary. Also the VLD criteria voltage can be set by the software. The criteria voltage can be set by matching to one of the 3 kinds of voltage 2.4V, 2.7V, 3.3V or 3.9V (VDD reference voltage).

The VLD block works only when VLDCON.0 is set. If VDD level is lower than the reference voltage selected with VLDCON.5-.1, VLDCON.6 will be set. If VDD level is higher, VLDCON.6 will be cleared. Please do not operate the VLD block for minimize power current consumption.

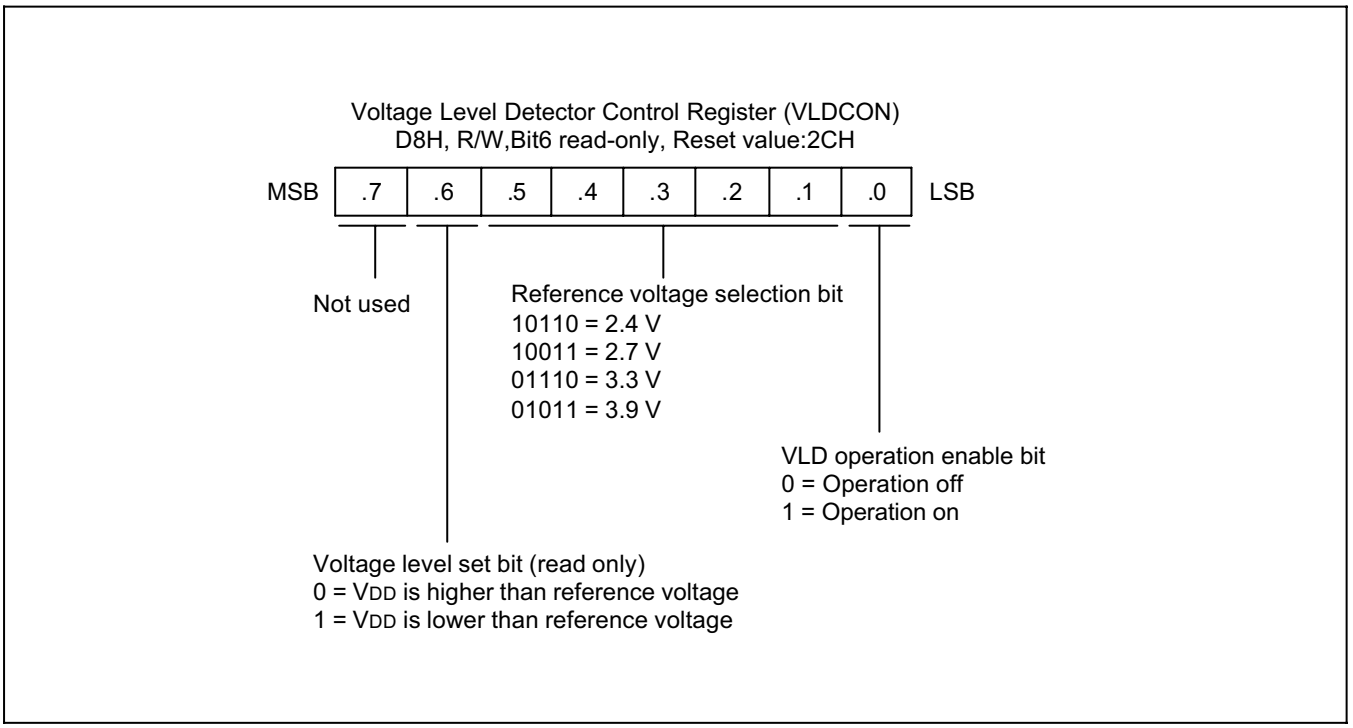


Figure 17-1. VLD Control Register (VLDCON)

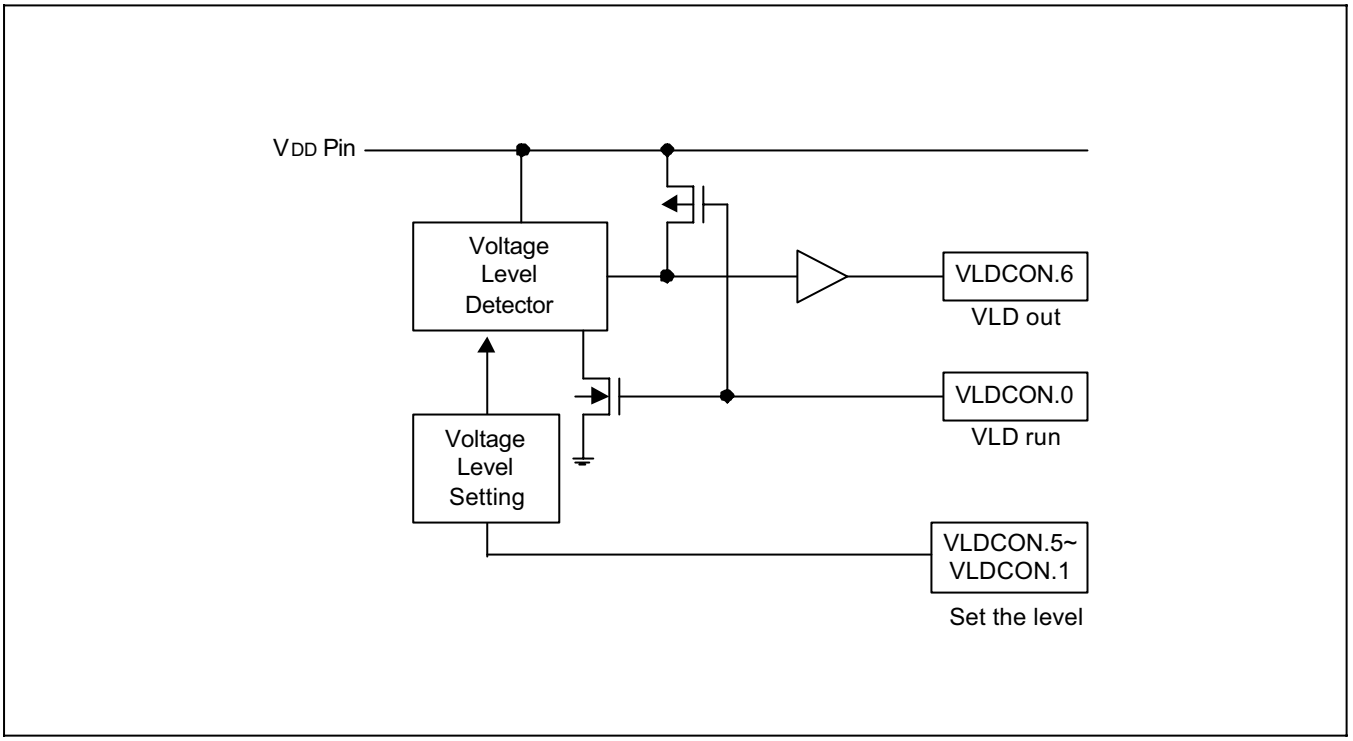


Figure 17-2. Block Diagram for Voltage Level Detect

VOLTAGE LEVEL DETECTOR CONTROL REGISTER (VLDCON)

The bit 0 of VLDCON controls to run or disable the operation of Voltage level detector. Basically this V_{VLD} is set as 2.4 V by system reset and it can be changed in 4 kinds voltages by selecting Voltage Level Detector Control register(VLDCON). When you write 5 bit data value to VLDCON, an established resistor string is selected and the V_{VLD} is fixed in accordance with this resistor. Table 17-1 shows specific V_{VLD} of 3 levels.

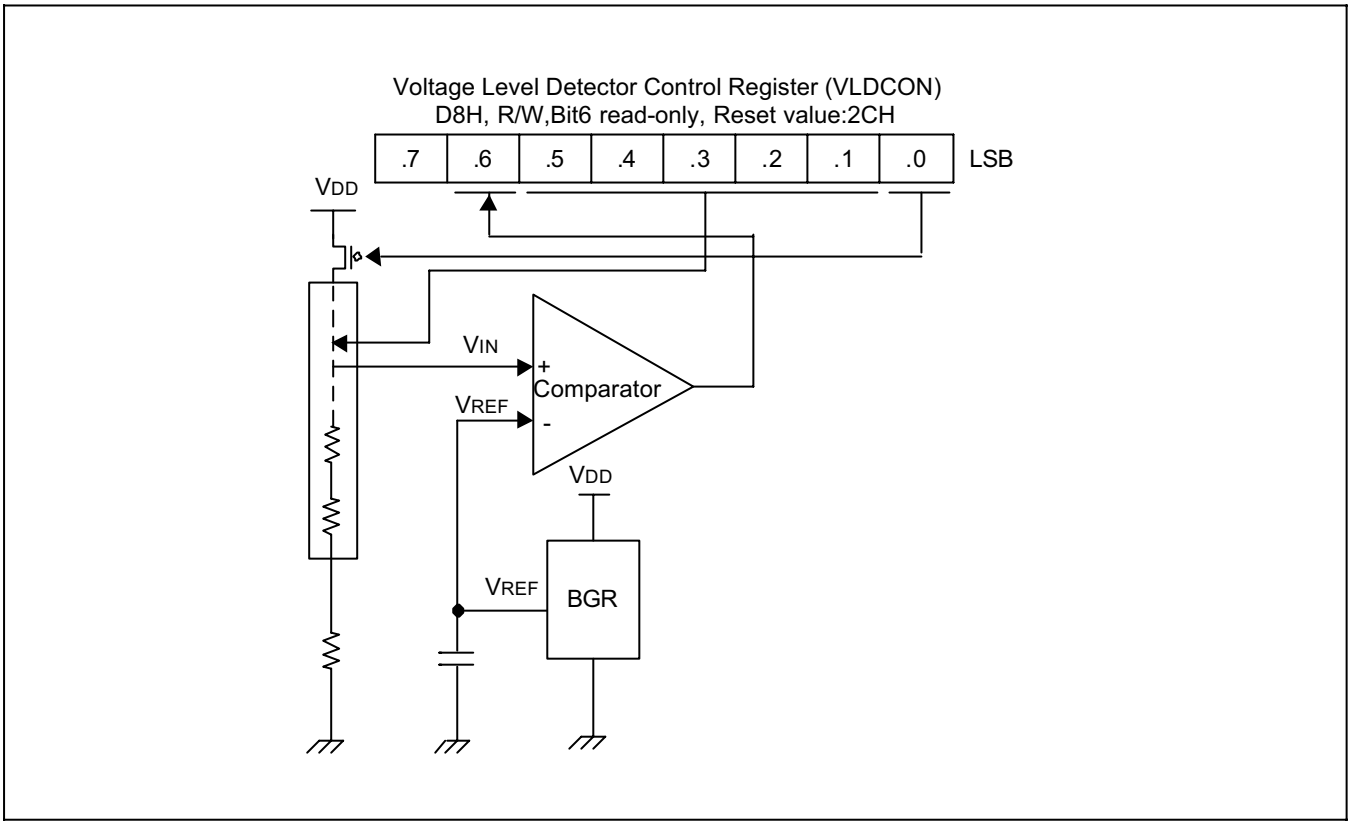


Figure 17-2. Voltage Level Detect Circuit and Control Register

Table 17-1. VLDCON Value and Detection Level

VLDCON .5-.1	V_{VLD}
10110	2.4 V
10011	2.7 V
01110	3.3 V
01011	3.9 V

NOTE: VLDCON reset value is 2CH .

VOLTAGE(VDD) LEVEL DETECTION SEQUENCE - VLD USAGE

STEP 0: Don't make VLD on in normal conditions for small current consumption.

STEP 1: For initializing analog comparator, write #3Fh to VLDCON. (Comparator initialization, VLD enable)

STEP 2: Write value to reference voltage setting bits in VLDCON. (Voltage setting, VLD enable)

STEP 3: Wait 10~20usec for comparator operation time. (Wait compare time)

STEP 4: Check result by loading voltage level set bit in VLDCON. (Check result)

STEP 5: For another measurement, repeat above steps.

PROGRAMING TIP

```
LD      VLDCON,#3FH          ;  Comparator initialization,VLD enable (STEP 1)

LD      VLDCON,#00011101B    ;  3.3V detection voltage setting, VLD enable (STEP 2)

NOP
NOP
NOP
•                            ;   Wait 10~20usec (STEP 3)
•
•
LD      R0, VLDCON           ;  Load VLDCON to R0 (STEP 4)
TM      R0, #01000000B       ;  Check bit6 of R0. If bit6 is "H", VDD is lower than 3.3V.
JP      NZ, LOW_VDD          ;   If not zero(bit 6 is "H"), jump to "LOW_VDD" routine.
```

Table 17-2. Characteristics of Voltage Level Detect Circuit (T_A = 25 °C)

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Operating Voltage	V_{DDVLD}		1.5	–	5.5	V
Detection Voltage	V_{VLD}	VLDCON.5–.1 = 10110b	2.0	2.4	2.8	
		VLDCON.5–.1 = 10011b	2.3	2.7	3.1	
		VLDCON.5–.1 = 01110b	2.9	3.3	3.7	
		VLDCON.5–.1 = 01011b	3.5	3.9	4.3	
Current consumption	I_{VLD}	VLD on V_{DD} = 5.5 V	–	65	100	uA
		V_{DD} = 3.0 V		45	80	

18 LOW VOLTAGE RESET

OVERVIEW

The S3C9484/C9488/F9488 can be reset in four ways:

— by external power-on-reset

— by the external reset input pin pulled low

— by the digital watchdog timing out

— by the Low Voltage reset circuit (LVR)

During an external power-on reset, the voltage VDD is High level and the RESETB pin is forced Low level. The RESETB signal is input through a Schmitt trigger circuit where it is then synchronized with the CPU clock. This brings the S3C9484/C9488/F9488 into a known operating status. To ensure correct start-up, the user should take that reset signal is not released before the VDD level is sufficient to allow MCU operation at the chosen frequency.

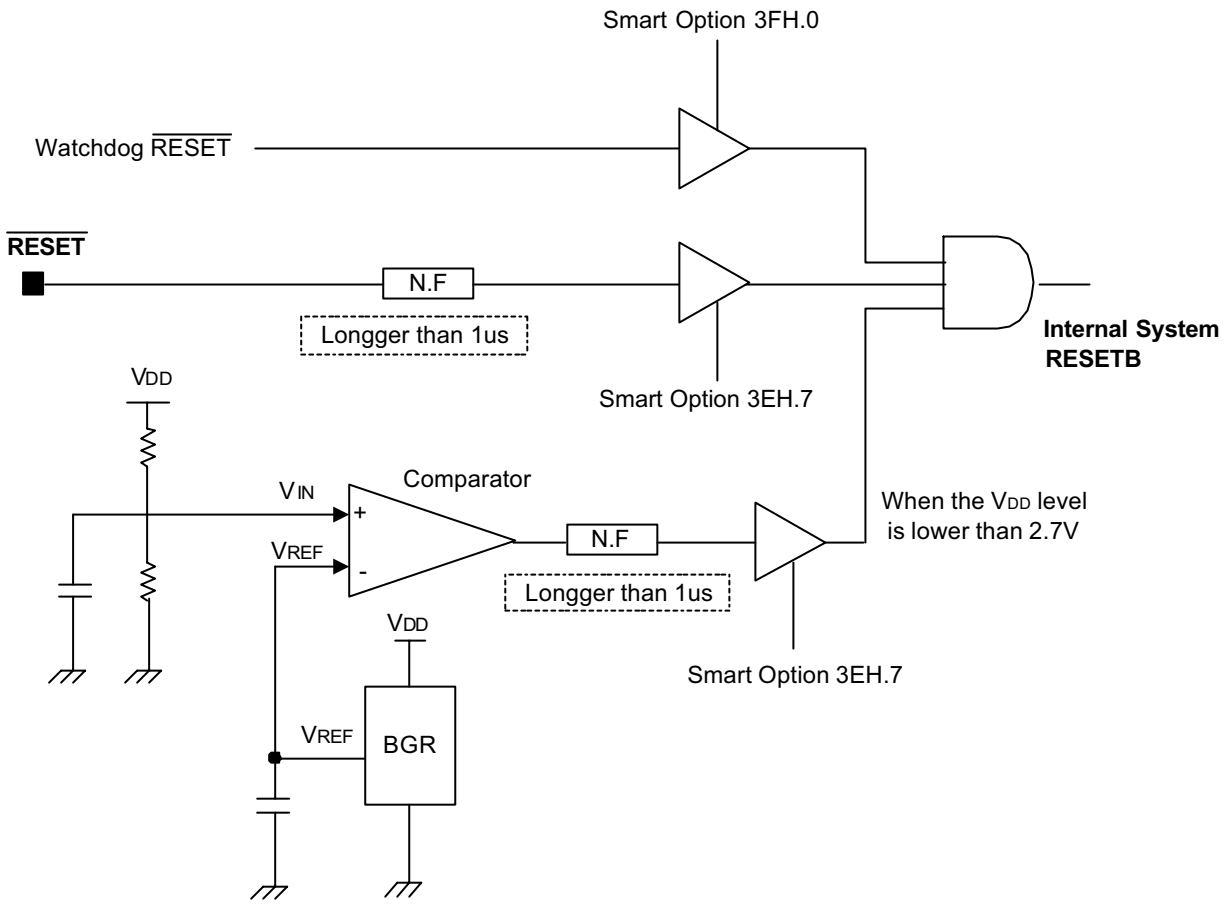
The RESETB pin must be held to Low level for a minimum time interval after the power supply comes within tolerance in order to allow time for internal CPU clock oscillation to stabilize. The minimum required oscillation stabilization time for a reset is approximately 8.19 ms ($\cong 2^{16}$/fosc, fosc= 8MHz).

When a reset occurs during normal operation (with both VDD and RESETB at High level), the signal at the RESETB pin is forced Low and the reset operation starts. All system and peripheral control registers are then set to their default hardware reset values (see Table 8-1).

The MCU provides a watchdog timer function in order to ensure graceful recovery from software malfunction. If watchdog timer is not refreshed before an end-of-counter condition (overflow) is reached, the internal reset will be activated.

The S3C9484/C9488/F9488 has a built-in low voltage reset circuit that allows detection of power voltage drop of external V_{DD} input level to prevent a MCU from malfunctioning in an unstable MCU power level. This voltage detector works for the reset operation of MCU. This Low Voltage reset includes an analog comparator and Vref circuit. The value of a detection voltage is set internally by hardware. The on-chip Low Voltage Reset, features static reset when supply voltage is below a reference voltage value (you did select at smart option 3FH). Thanks to this feature, external reset circuit can be removed while keeping the application safety. As long as the supply voltage is below the reference value, there is an internal and static RESET. The MCU can start only when the supply voltage rises over the reference voltage.

When you calculate power consumption, please remember that a static current of LVR circuit should be added a CPU operating current in any operating modes such as Stop, Idle, and normal RUN mode.



NOTES:
1. The target of voltage detection level is that you did select at smart option 3EH.
2. BGR is Band Gap voltage Reference

Figure 18-1. Low Voltage Reset Circuit

NOTE

To program the duration of the oscillation stabilization interval, you make the appropriate settings to the basic timer control register, BTCON, *before* entering Stop mode. Also, if you do not want to use the watchdog function (which causes a system reset if a watchdog timer counter overflow occurs), you can disable it by writing '1010B' to the upper nibble of WDTCON.

19 ELECTRICAL DATA

OVERVIEW

In this chapter, S3C9484/C9488/F9488 electrical characteristics are presented in tables and graphs. The information is arranged in the following order:

— Absolute maximum ratings
— Input/output capacitance
— D.C. electrical characteristics
— A.C. electrical characteristics
— Oscillation characteristics
— Oscillation stabilization time
— Data retention supply voltage in stop mode
— A/D converter electrical characteristics

Table 19-1. Absolute Maximum Ratings

(T_A = 25 °C)

Parameter	Symbol	Conditions	Rating	Unit
Supply voltage	V_{DD}		– 0.3 to +6.5	V
Input voltage	V_I		– 0.3 to V_{DD} + 0.3	
Output voltage	V_O		– 0.3 to V_{DD} + 0.3	
Output current high	I_{OH}	One I/O pin active	– 18	mA
		All I/O pins active	– 60	
Output current low	I_{OL}	One I/O pin active	+30	
		Total pin current for port	+100	
Operating temperature	T_A		– 25 to + 85	°C
Storage temperature	T_{STG}		– 65 to + 150	

Table 19-2. D.C. Electrical Characteristics

(T_A = -25 °C to + 85 °C, V_{DD} = 2.2 V to 5.5 V)

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Operating voltage	V_{DD}	f_{CPU} = 8 MHz	2.7	–	5.5	V
		f_{CPU} = 4 MHz	2.2		5.5	
Input high voltage	V_{IH1}	All input pins except V_{IH2}	0.8 V_{DD}		V_{DD}	
	V_{IH2}	X_{IN}, XT_{IN}	V_{DD}-0.5			
Input low voltage	V_{IL1}	All input pins except V_{IL2}	–		0.2 V_{DD}	
	V_{IL2}	X_{IN}, XT_{IN}			0.5	

Table 19-2. D.C. Electrical Characteristics (Continued)

(T_A = -25 °C to + 85 °C, V_{DD} = 2.2 V to 5.5 V)

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Output high voltage	V_{OH1}	V_{DD} = 2.4 V; I_{OH} = -4 mA P1.0-P1.1 and P3.4-P3.6	V_{DD} - 0.7	V_{DD} - 0.3	–	V
	V_{OH2}	V_{DD} = 5 V; I_{OH} = -4 mA Port 2	V_{DD} - 1.0	–	–	
	V_{OH3}	V_{DD} = 5 V; I_{OH} = -1 mA Normal output pins	V_{DD} - 1.0	–	–	
Output low voltage	V_{OL1}	V_{DD} = 2.4 V; I_{OL} = 12 mA P1.0-P1.1 and P3.4-P3.6		0.3	0.5	
	V_{OL2}	V_{DD} = 5 V; I_{OL} = 15 mA Port 2		0.4	2.0	
	V_{OL3}	V_{DD} = 5 V; I_{OL} = 4 mA Normal output pins	–	0.4	2.0	
Input high leakage current	I_{LIH1}	V_{IN} = V_{DD} All input pins except I_{LIH2}	–	–	3	μA
	I_{LIH2}	V_{IN} = V_{DD}, X_{IN}, XT_{IN}			20	
Input low leakage current	I_{LIL1}	V_{IN} = 0 V All input pins except I_{LIL2}	–	–	-3	
	I_{LIL2}	V_{IN} = 0 V, X_{IN}, XT_{IN}			-20	
Output high leakage current	I_{LOH}	V_{OUT} = V_{DD} All I/O pins and Output pins	–	–	3	
Output low leakage current	I_{LOL}	V_{OUT} = 0 V All I/O pins and Output pins	–	–	-3	
Oscillator feed back resistors	R_{OSC1}	V_{DD} = 5.0 V, T_A = 25 °C X_{IN} = V_{DD}, X_{OUT} = 0 V	800	1000	1200	kΩ
Pull-up resistor	R_{L1}	V_{IN} = 0 V; V_{DD} = 5 V ±10 % Port 0,1,2,3,4 T_A = 25°C	25	50	100	
COM output voltage deviation	V_{DC}	V_{DD} = V_{LC4} = 5 V (V_{LC4}-COMi) IO = ± 15 p-μA (i = 0-7)	–	± 45	± 90	mV
SEG output voltage deviation	V_{DS}	V_{DD} = V_{LC4} = 5 V (V_{LC4}-SEGi) IO = ± 15 p-μA (i = 0-18)	–	± 45	± 90	

Table 19-2. D.C. Electrical Characteristics (Concluded)

(T_A = -25 °C to + 85 °C, V_{DD} = 2.2V to 5.5 V)

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
LCD Voltage Dividing Resister	R_{LCD}	–	40	75	100	kΩ
V_{LC3} OUTPUT VOLTAGE	V_{LC3}	V_{DD}=1.8V to 5.5V, 1/4 bias LCD clock=0Hz, V_{LC4}=V_{DD}	0.75V_{DD}-0.2	0.75V_{DD}	0.75V_{DD}+0.2	V
V_{LC2} OUTPUT VOLTAGE	V_{LC2}		0.5V_{DD}-0.2	0.5V_{DD}	0.5V_{DD}+0.2	V
V_{LC1} OUTPUT VOLTAGE	V_{LC1}		0.25V_{DD}-0.2	0.25V_{DD}	0.25V_{DD}+0.2	V
Supply current (1)	I_{DD1} (2)	V_{DD} = 5 V ± 10 % 8 MHz crystal oscillator	–	12	25	mA
		4 MHz crystal oscillator		4	10	
		V_{DD} = 3 V ± 10 % 8 MHz crystal oscillator		3	8	
		4 MHz crystal oscillator		1	5	
	I_{DD2}	Idle mode: V_{DD} = 5 V ± 10 % 8 MHz crystal oscillator		3	10	
		4 MHz crystal oscillator		1.5	4	
		Idle mode: V_{DD} = 3 V± 10 % 8 MHz crystal oscillator		1.2	3	
		4 MHz crystal oscillator		1.0	2.0	
	I_{DD3}	Sub operating: main-osc stop V_{DD} = 3 V ± 10 % 32768 Hz crystal oscillator		40	80	μA
	I_{DD4}	Sub idle mode: main osc stop V_{DD} = 3 V ± 10 % 32768 Hz crystal oscillator		7	14	
	I_{DD5}	Main stop mode : sub-osc stop V_{DD} = 5 V ± 10 %, T_A = 25 °C		1	3	
		V_{DD} = 3 V ± 10 %, T_A = 25 °C		0.5	2	

NOTES:

1. Supply current does not include current drawn through internal pull-up resistors or external output current loads.
2. I_{DD1} and I_{DD2} include a power consumption of subsystem oscillator.
3. I_{DD3} and I_{DD4} are the current when the main system clock oscillation stop and the subsystem clock is used. And they does not include the LCD and Voltage booster and voltage level detector current.
4. I_{DD5} is the current when the main and subsystem clock oscillation stop.
5. Voltage booster's operating voltage rage is 2.0V to 5.5V.
6. If you use LVR module, supply current increase. (refer to Table 19-12)

Table 19-3. A.C. Electrical Characteristics

(T_A = -25 °C to +85 °C, V_{DD} = 2.2 V to 5.5 V)

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Interrupt input high, low width (P3.3–P3.6)	t_{INTH}, t_{INTL}	P3.3–P3.6, V_{DD} = 5 V	200	–	–	ns
RESET input low width	t_{RSL}	V_{DD} = 5 V	1.5	–	–	µS

NOTE: User must keep more large value then min value.

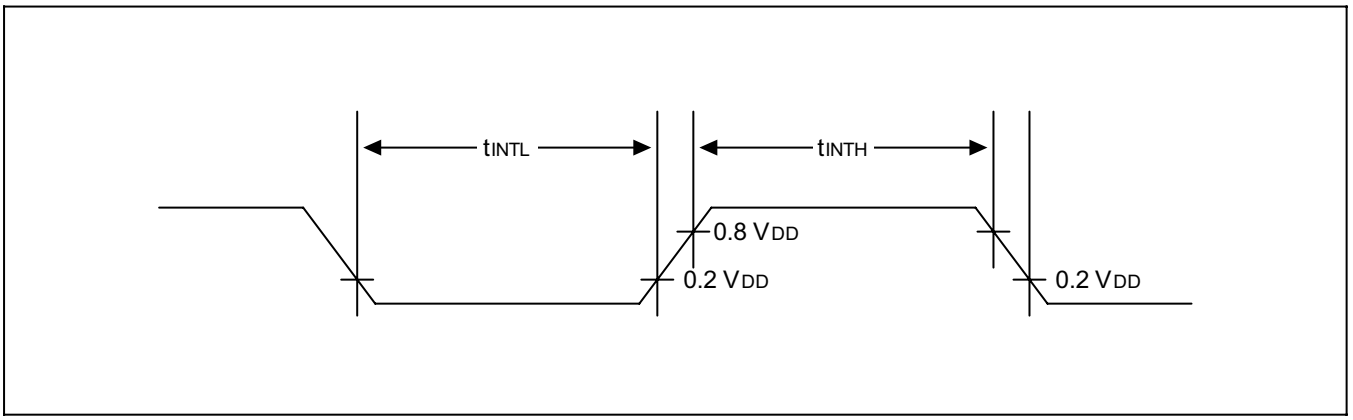


Figure 19-1. Input Timing for External Interrupts (P3.3–P3.6)

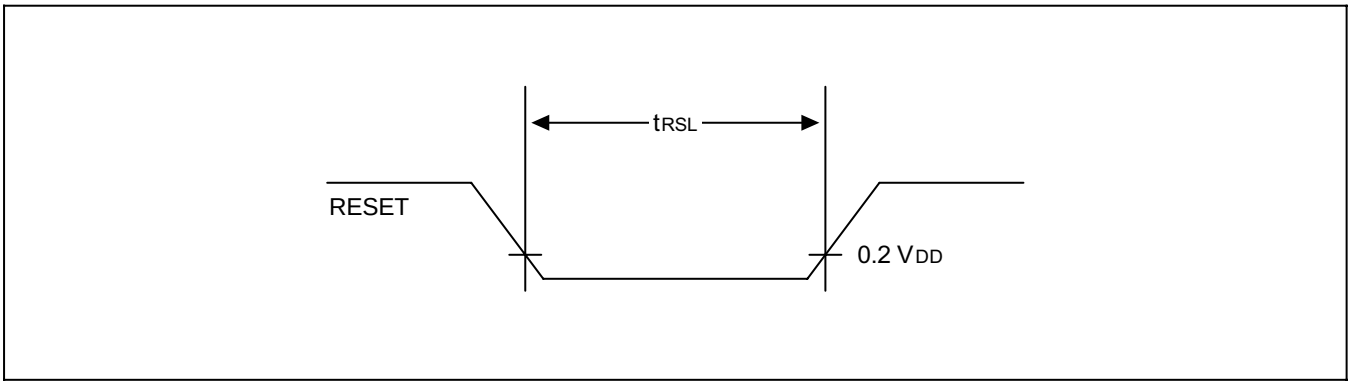


Figure 19-2. Input Timing for RESET

Table 19-4. Input/Output Capacitance

(T_A = -25 °C to +85 °C, V_{DD} = 0 V)

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Input capacitance	C_{IN}	f = 1 MHz; unmeasured pins are returned to V_{SS}	–	–	10	pF
Output capacitance	C_{OUT}					
I/O capacitance	C_{IO}					

Table 19-5. Data Retention Supply Voltage in Stop Mode

(T_A = -25 °C to + 85 °C)

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Data retention supply voltage	V_{DDDR}		2	–	5.5	V
Data retention supply current	I_{DDDR}	V_{DDDR} = 2 V	–	–	3	µA

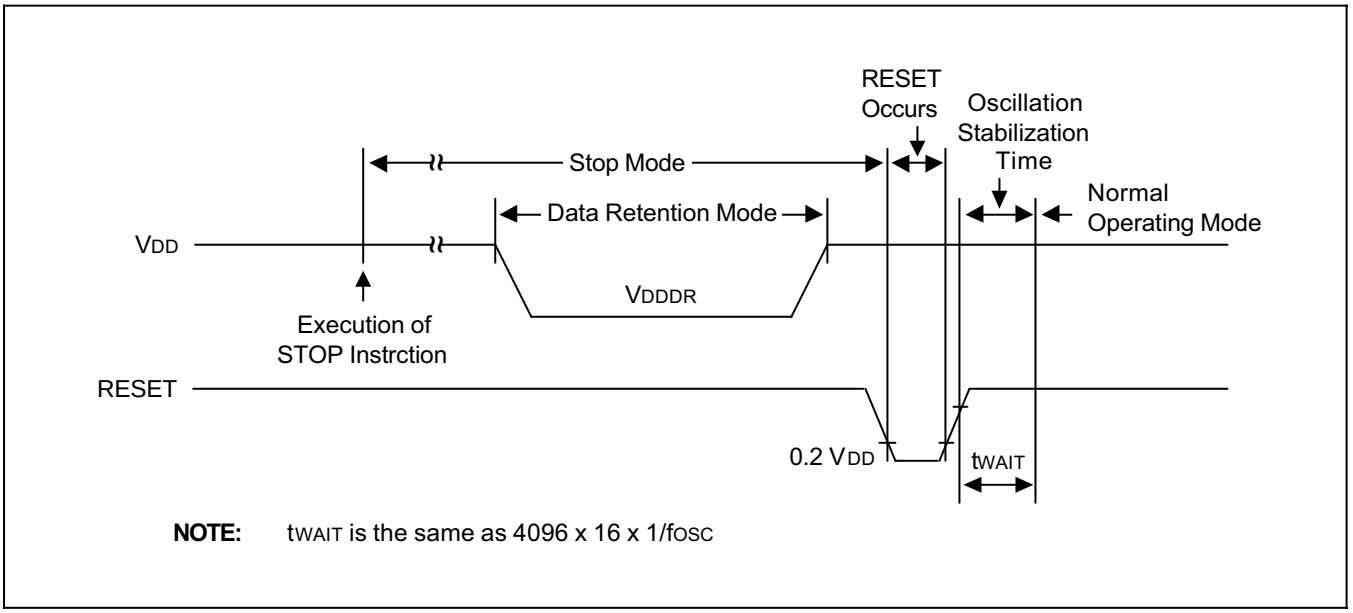


Figure 19-3. Stop Mode Release Timing Initiated by RESET

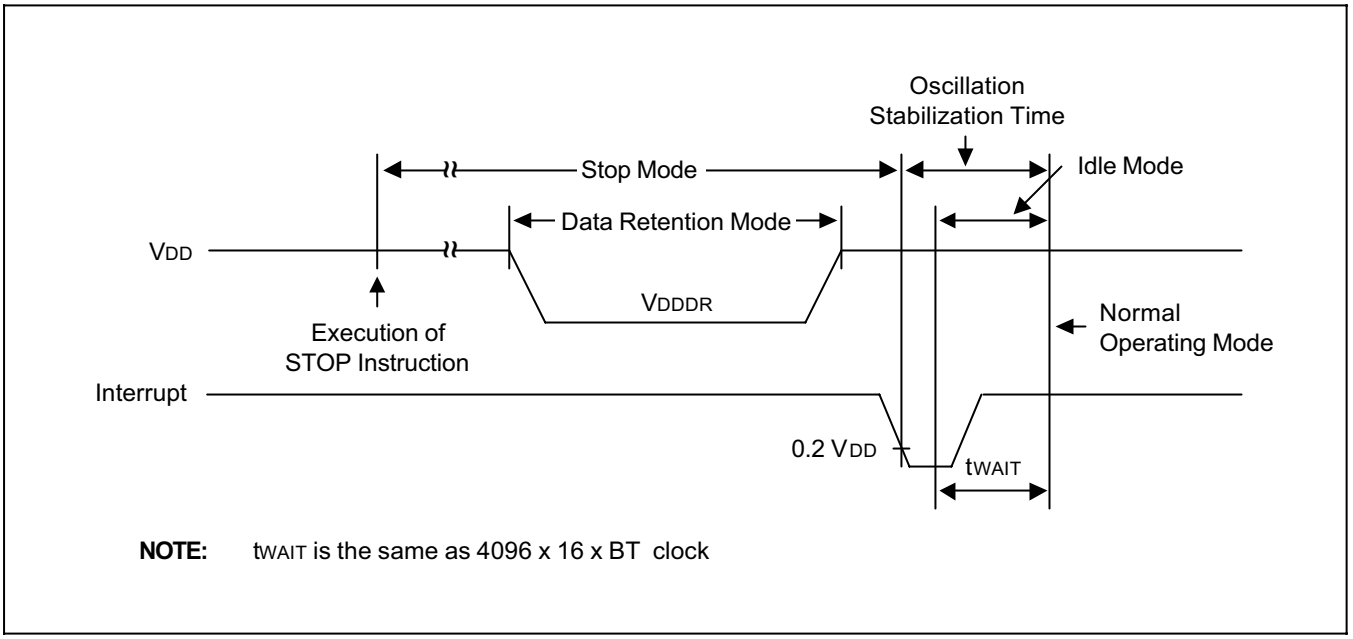


Figure 19-4. Stop Mode(main) Release Timing Initiated by Interrupts

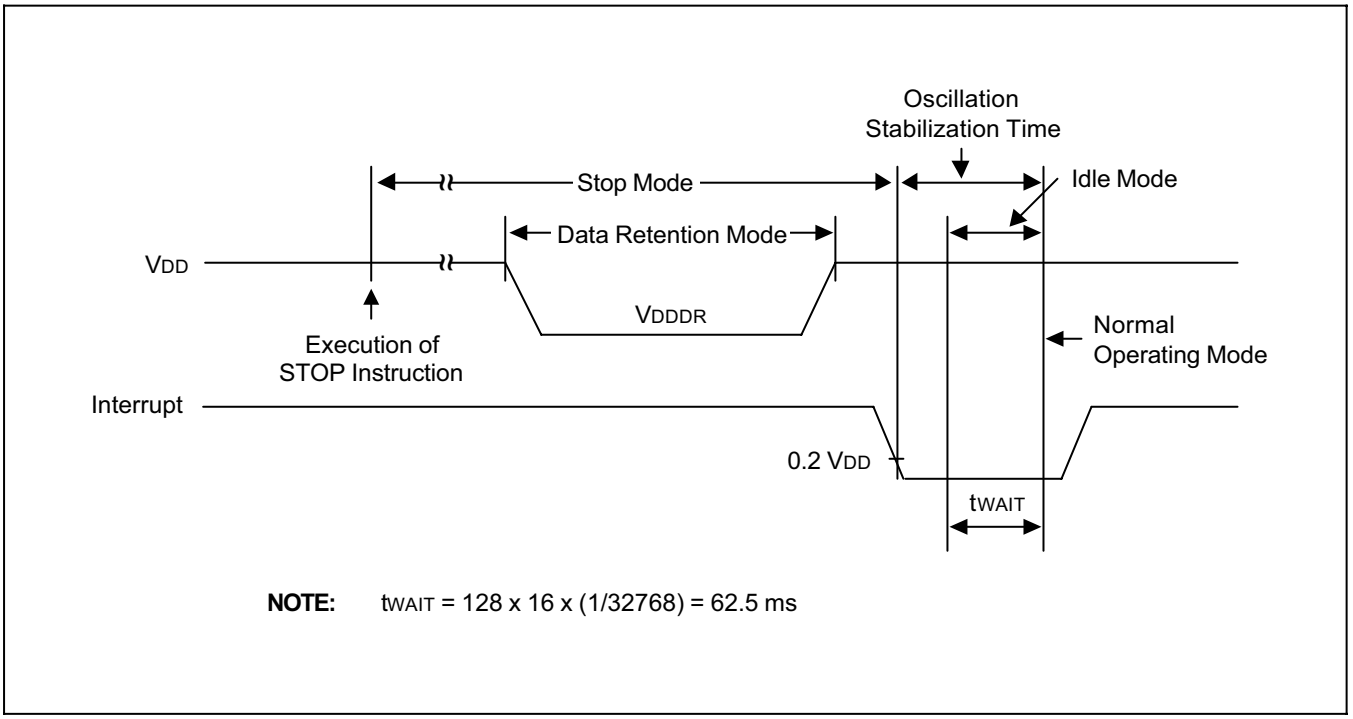


Figure 19-5. Stop Mode(sub) Release Timing Initiated by Interrupts

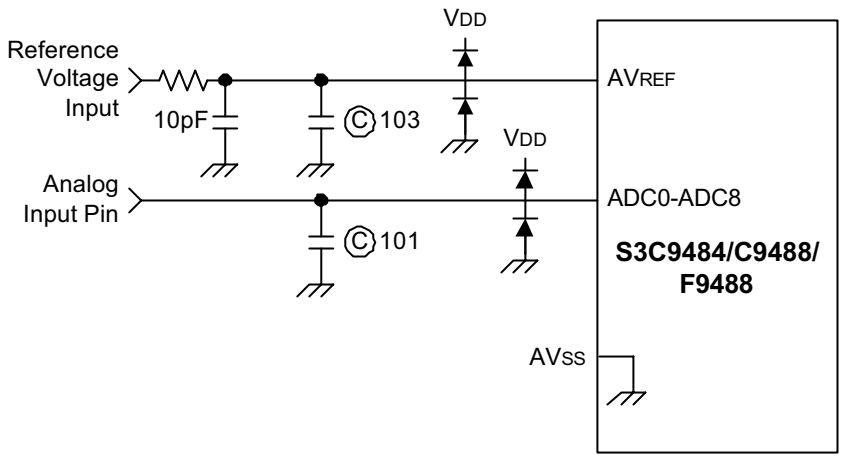
Table 19-6. A/D Converter Electrical Characteristics

(T_A = - 25 °C to +85 °C, V_{DD} = 2.2 V to 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Resolution			–	10	–	bit
Total accuracy		V_{DD} = 5.12 V	–	–	±3	LSB
Integral Linearity Error	ILE	AV_{REF} = 5.12V	–	–	±3	
Differential Linearity Error	DLE	AV_{SS} = 0 V CPU clock = 8 MHz	–	–	±1	
Offset Error of Top	EOT		–	±1	±3	
Offset Error of Bottom	EOB		–	±1	±3	
Conversion time (1)	T_{CON}	10-bit resolution 50 x fxx/4, fxx = 8MHz	20	–	–	µS
Analog input voltage	V_{IAN}	–	AV_{SS}	–	AV_{REF}	V
Analog input impedance	R_{AN}	–	2	1000	–	MΩ
Analog reference voltage	AV_{REF}	–	2.5	–	V_{DD}	V
Analog ground	AV_{SS}	–	V_{SS}	–	V_{SS} +0.3	
Analog input current	I_{ADIN}	AV_{REF} = V_{DD} = 5V	–	–	10	µA
Analog block current (2)	I_{ADC}	AV_{REF} = V_{DD} = 5V	–	1	3	mA
		AV_{REF} = V_{DD} = 3V		0.5	1.5	
		AV_{REF} = V_{DD} = 5V When Power Down mode		100	500	nA

NOTES:

1. 'Conversion time' is the time required from the moment a conversion operation starts until it ends.
2. I_{ADC} is an operating current during A/D conversion.

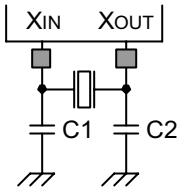
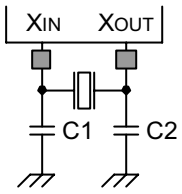
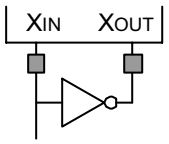
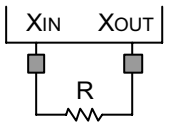


NOTE: The symbol 'R' signifies an offset resistor with a value of from 50 to 100.
If this resistor is omitted, the absolute accuracy will be maximum of 3 LSBs.

Figure 19-6. Recommended A/D Converter Circuit for Highest Absolute Accuracy

Table 19-7. Main Oscillator Frequency (f_{OSC1})

(T_A = -25 °C to +85 °C, V_{DD} = 2.2 V to 5.5 V)

Oscillator	Clock Circuit	Test Condition	Min	Typ	Max	Unit
Crystal		(1) Crystal oscillation Frequency (2) Crystal = 8MHz C1 = 20 pF, C2 = 20 pF	1	–	8	MHz
Ceramic		Ceramic oscillation frequency	1	–	8	
External clock		X_{IN} input frequency	1	–	8	
RC		r = 35 KΩ, V_{DD} = 5 V		2		

NOTES:
1. We recommend crystal of TDK Korea as the most suitable oscillator of Samsung Microcontroller. If you want to know detailed information of Crystal Oscillator Frequency with cap, please visit the web site(www.tdkkorea.co.kr).
2. The value of Crystal(10MHz) and Cap(20pF) is based on TDK Korea parts.

Table 19-8. Main Oscillator Clock Stabilization Time (t_{ST1})

(T_A = -25 °C to +85 °C, V_{DD} = 2.2V to 5.5 V)

Oscillator	Test Condition	Min	Typ	Max	Unit
Crystal	V_{DD} = 4.5 V to 5.5 V	–	–	10	ms
	V_{DD} = 2.2 V to 4.5 V			30	
Ceramic	Stabilization occurs when V_{DD} is equal to the minimum oscillator voltage range.	–	–	4	
External clock	X_{IN} input high and low level width (t_{XH}, t_{XL})	50	–	–	ns

NOTE: Oscillation stabilization time (t_{ST1}) is the time required for the CPU clock to return to its normal oscillation frequency after a power-on occurs, or when Stop mode is ended by a RESET signal.

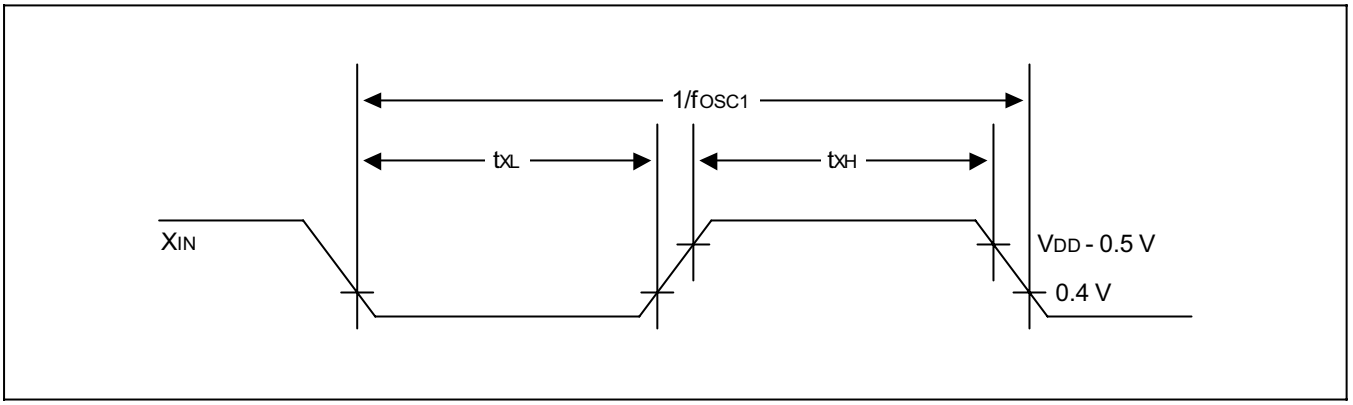


Figure 19-7. Clock Timing Measurement at X_{IN}

Table 19-9. Sub Oscillator Frequency (f_{OSC2})

(T_A = -25 °C + 85 °C, V_{DD} = 2.2 V to 5.5 V)

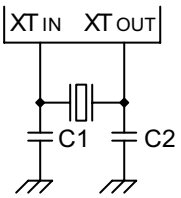
Oscillator	Clock Circuit	Test Condition	Min	Typ	Max	Unit
Crystal		C1 = 33 pF, C2 = 33 pF	32	32.768	35	kHz

Table 19-10. Sub Oscillator(crystal) Stabilization Time (t_{ST2})

(T_A = 25 °C, V_{DD} = 2.2 V to 5.5 V))

Test Condition	Min	Typ	Max	Unit
V_{DD} = 4.5 V to 5.5 V	–	250	500	ms
V_{DD} = 2.2 V to 4.5 V	–	–	10	s

NOTE: Oscillation stabilization time (t_{ST2}) is the time required for the CPU return to its normal operation when Stop mode is released by interrupts.

Table 19-11. LCD Contrast Controller Characteristics

(T_A = – 25 °C to + 85 °C, V_{DD} = 4.5 V to 5.5 V)

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Resolution	–	–	–	–	4	Bits
Linearity	RLIN	–	–	–	± 1.0	LSB
Max Output Voltage (LCDVOL=#8FH)	VLPP	VLC4=V_{DD}=5V	4.9	–	VLC1	V

Table 19-12. LVR (Low Voltage Reset) Circuit Characteristics

(T_A = 25 °C)

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
LVR Voltage high	V_{LVRH}		2.8 3.5 4.1			V
LVR Voltage low	V_{LVRL}		2.4 3.1 3.7	2.6 3.3 3.9	2.8 3.5 4.1	
Power supply voltage rising time	T_R		10			µS
Power supply voltage off time	T_{OFF}		0.5			S
LVR circuit consumption current	I_{DDPR}	V_{DD} = 5V +/- 10%		65	100	µA
		V_{DD} = 3V		45	80	

NOTES:

1. 2^{16}/fx (= 8.19ms at fx = 8 MHz)
2. Current consumed when Low Voltage reset circuit is provided internally.

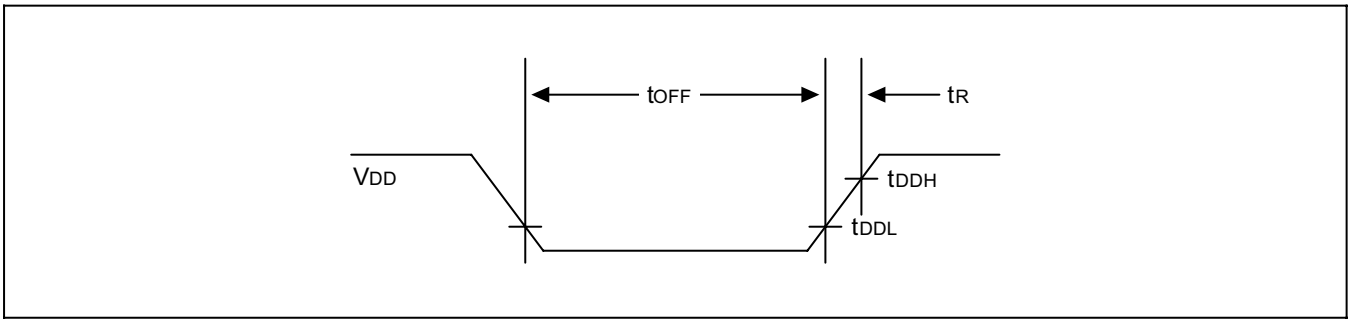


Figure 19-8. LVR (Low Voltage Reset) Timing

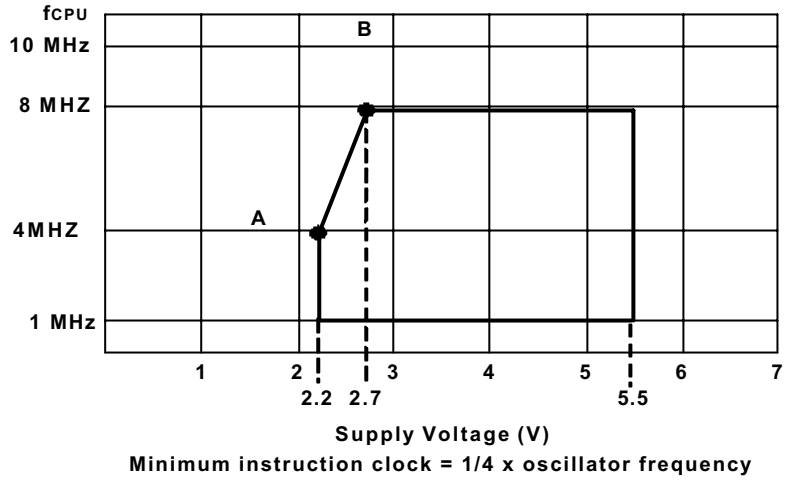


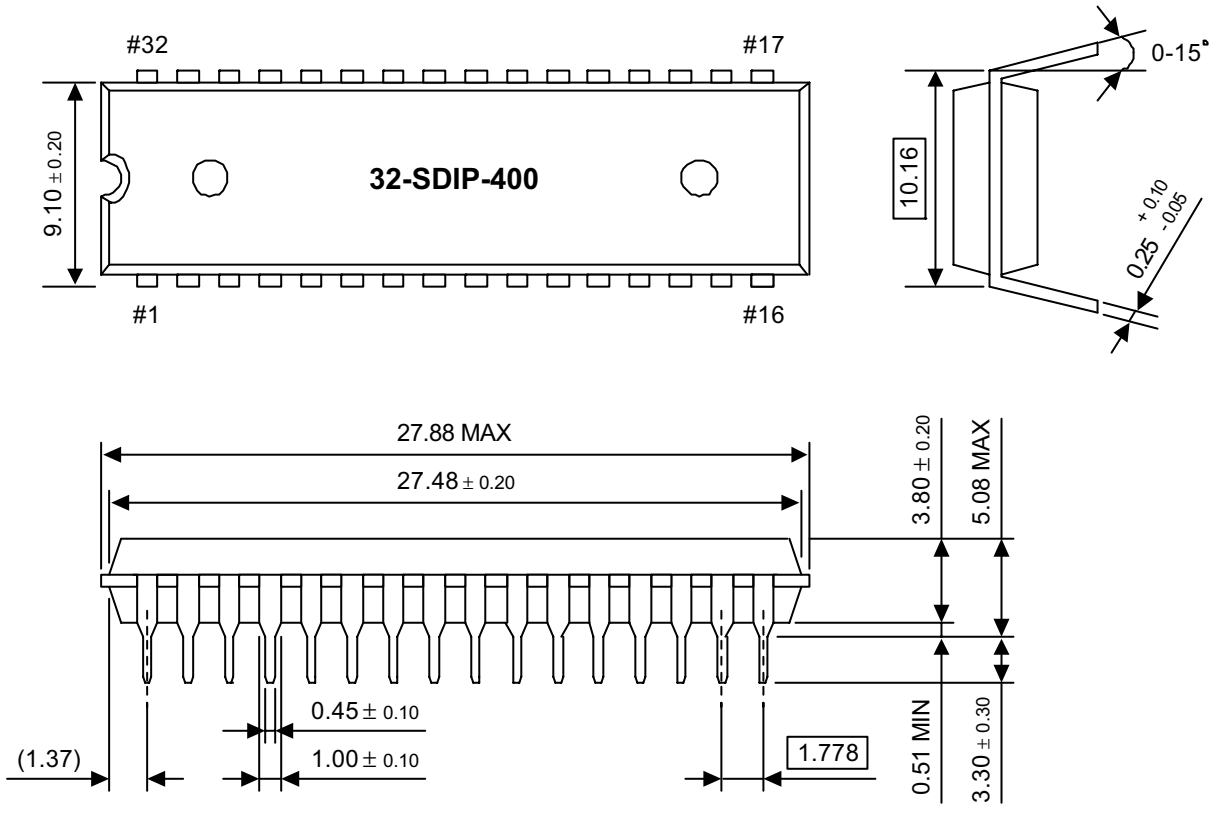
Figure 19-9. Operating Voltage Range

NOTES

20 MECHANICAL DATA

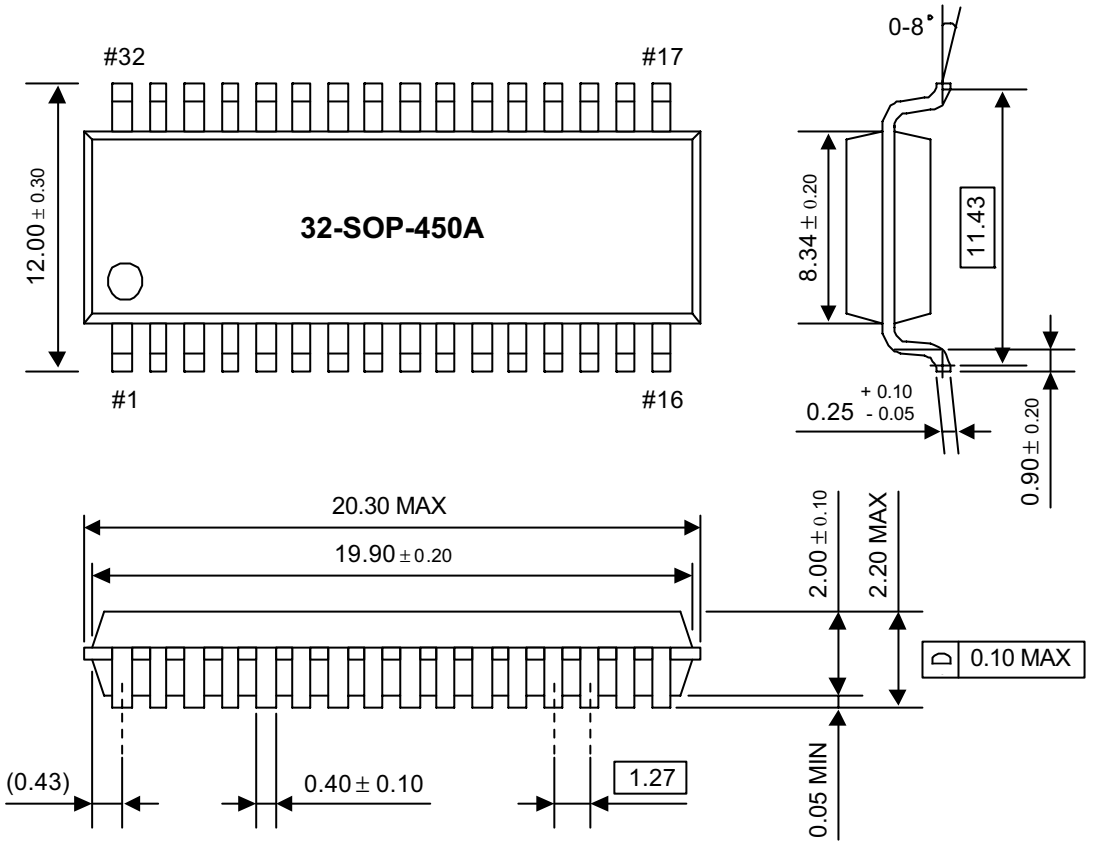
OVERVIEW

The S3C9484/C9488/F9488 microcontroller is currently available in 32-SDIP, 32-SOP, 42-SDIP, 44-QFP package.



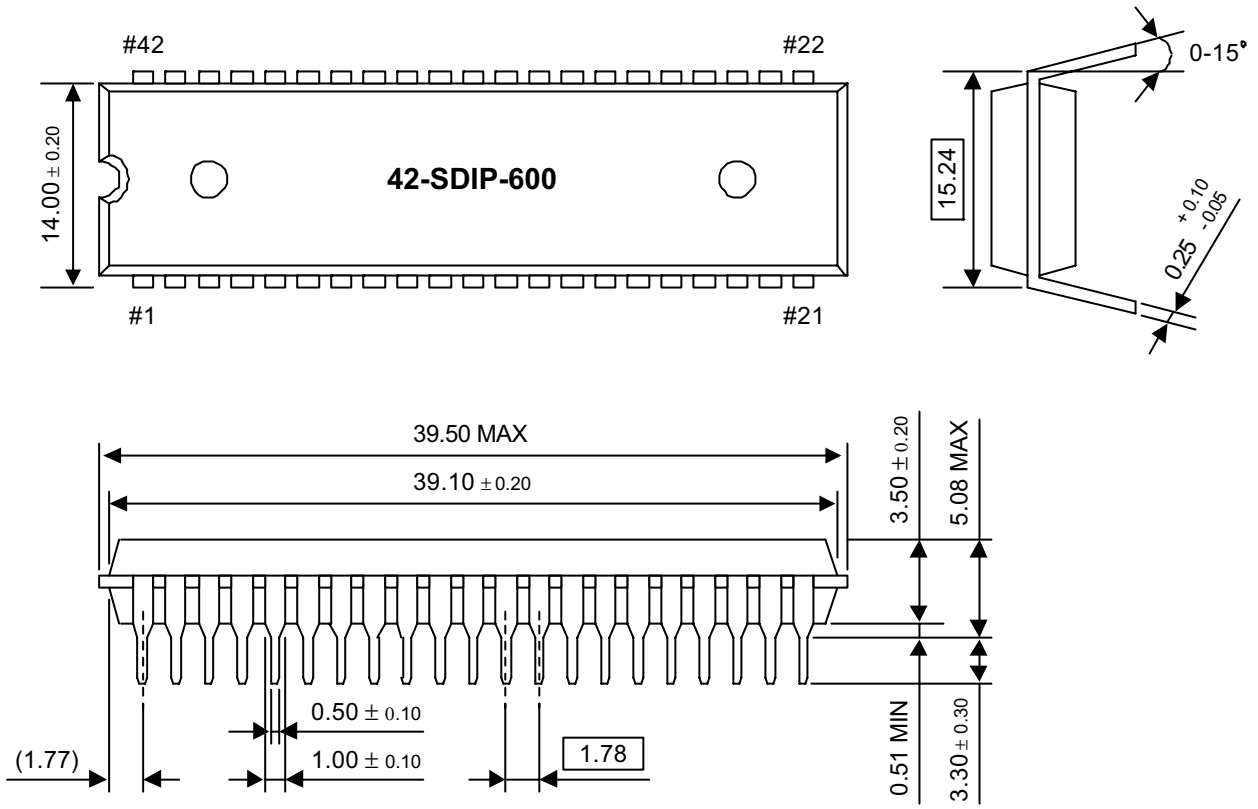
NOTE: Dimensions are in millimeters.

Figure 20-1. 32-SDIP-400 Package Dimensions



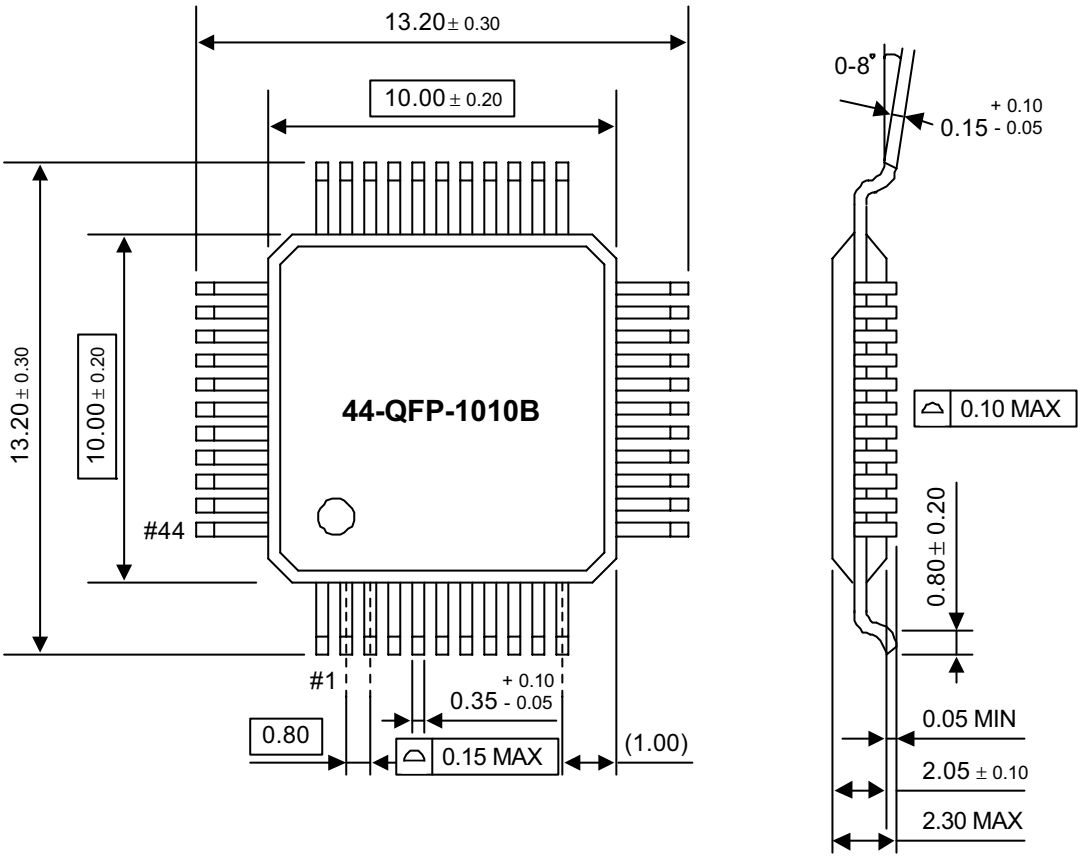
NOTE: Dimensions are in millimeters.

Figure 20-2. 32-SOP-450A Package Dimensions



NOTE: Dimensions are in millimeters.

Figure 20-3. 42-SDIP-600 Package Dimensions



NOTE: Dimensions are in millimeters.

Figure 20-4. 44-QFP-1010 Package Dimensions

21 MTP

OVERVIEW

The S3F9488 single-chip CMOS microcontroller is the MTP (Multi Time Programmable) version of the S3C9484/C9488 microcontroller. It has an on-chip Half Flash ROM instead of masked ROM. The Half Flash ROM is accessed by serial data format. The Half Flash ROM can be rewritten up to 100 times.

The S3F9488 is fully compatible with the S3C9484/C9488, in function, in D.C. electrical characteristics, and in pin configuration. Because of its simple programming requirements, the S3F9488 is ideal for use as an evaluation chip for the S3C9484/C9488.

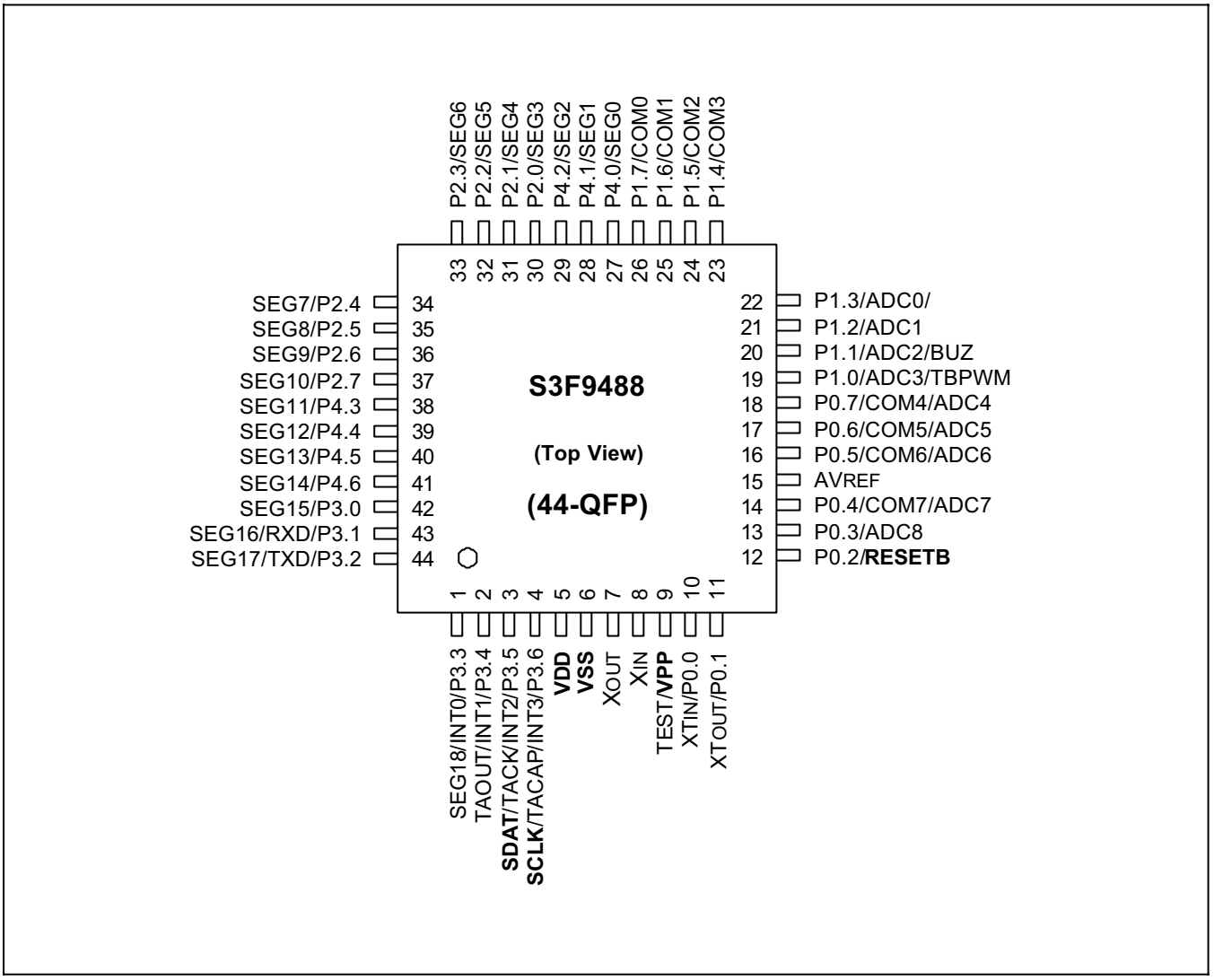


Figure 21-1. Pin Assignment Diagram (44-Pin Package)

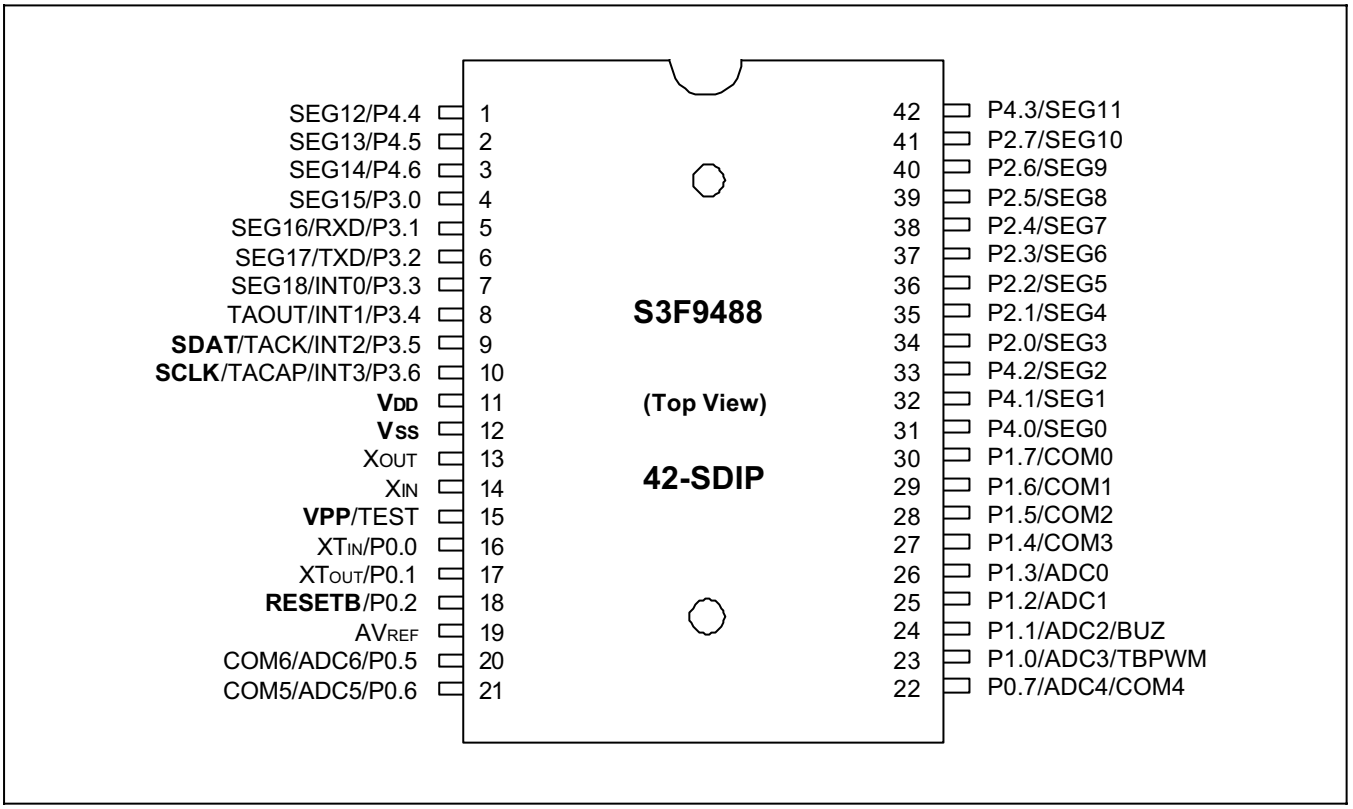


Figure 21-2. Pin Assignment Diagram (42-Pin Package)

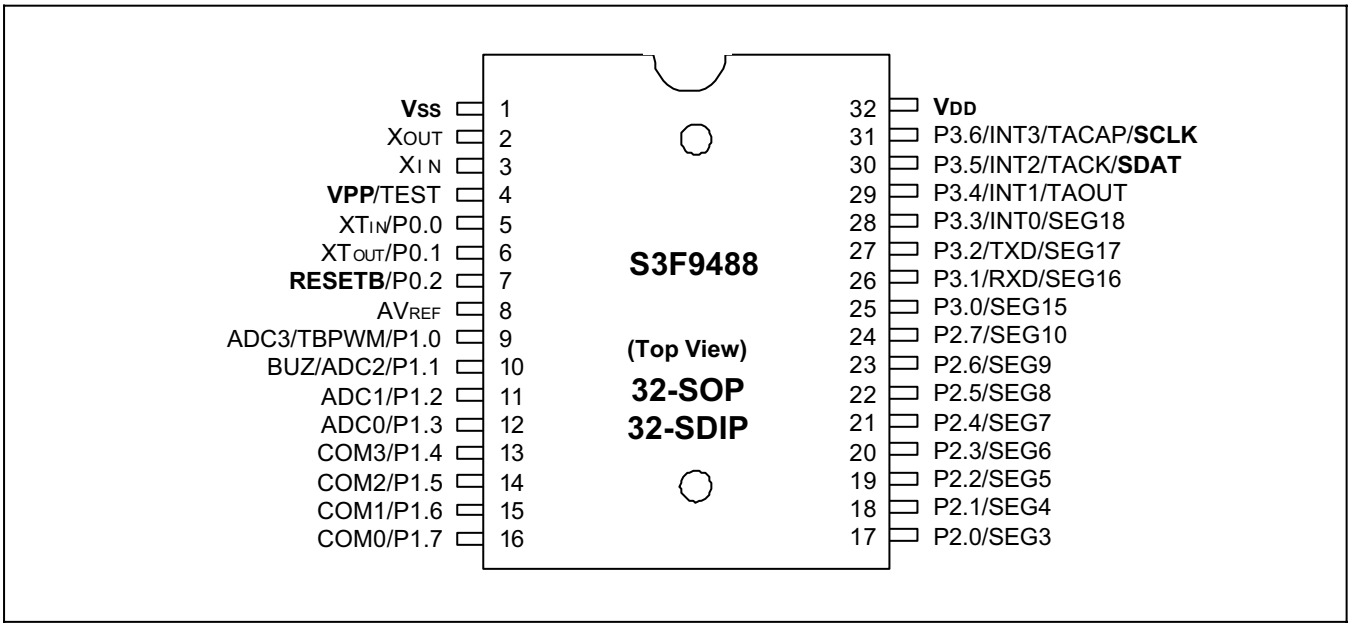


Figure 21-3. Pin Assignment Diagram (32-Pin Package)

Table 21-1. Descriptions of Pins Used to Read/Write the Flash ROM

Main Chip Pin Name	During Programming			
	Pin Name	Pin No.	I/O	Function
P3.5	SDAT	3 (44-pin) 9 (42-pin) 30 (32-pin)	I/O	Serial data pin (output when reading, Input when writing) Input and push-pull output port can be assigned
P3.6	SCLK	4 (44-pin) 10 (42-pin) 31 (32-pin)	I	Serial clock pin (input only pin)
TEST	VPP	9 (44-pin) 15 (42-pin) 4 (32-pin)	I	Power supply pin for flash ROM cell writing (indicates that MTP enters into the writing mode). When 12.5 V is applied, MTP is in writing mode and when 5 V is applied, MTP is in reading mode. (Option)
P0.2	RESETB	12 (44-pin) 18 (42-pin) 7 (32-pin)	I	
V_{DD}/V_{SS}	V_{DD}/V_{SS}	5/6 (44-pin) 11/12 (42-pin) 32/1 (32-pin)	I	Logic power supply pin.

Table 21-2. Comparison of S3F9488 and S3C9484/C9488 Features

Characteristic	S3F9488	S3C9484/C9488
Program Memory	8 Kbyte Flash ROM	4K/8K byte mask ROM
Operating Voltage (V_{DD})	2.2(2.7) V to 5.5 V	2.2(2.7) V to 5.5 V
MTP Programming Mode	VDD = 5 V, VPP = 12.5 V	
Pin Configuration	44QFP / 42SDIP / 32SDIP/ 32SOP	
EPROM Programmability	User Program multi time	Programmed at the factory

22 DEVELOPMENT TOOLS

OVERVIEW

Samsung provides a powerful and easy-to-use development support system on a turnkey basis. The development support system is composed of a host system, debugging tools, and supporting software. For a host system, any standard computer that employs Win95/98/2000/XP as its operating system can be used. A sophisticated debugging tool is provided both in hardware and software: the powerful in-circuit emulator, SMDS2+ or SK-1000, for the S3C7-, S3C9-, and S3C8- microcontroller families. SMDS2+ is a newly improved version of SMDS2, and SK-1000 is supported by a third party tool vendor. Samsung also offers supporting software that includes, debugger, an assembler, and a program for setting options.

SHINE

Samsung Host Interface for In-Circuit Emulator, SHINE, is a multi-window based debugger for SMDS2+. SHINE provides pull-down and pop-up menus, mouse support, function/hot keys, and context-sensitive hyper-linked help. It has an advanced, multiple-windowed user interface that emphasizes ease of use. Each window can be easily sized, moved, scrolled, highlighted, added, or removed.

SASM

The SASM takes a source file containing assembly language statements and translates them into a corresponding source code, an object code and comments. The SASM supports macros and conditional assembly. It runs on the MS-DOS operating system. As it produces the re-locatable object codes only, the user should link object files. Object files can be linked with other object files and loaded into memory. SASM requires a source file and an auxiliary register file (device_name.reg) with device specific information.

SAMA ASSEMBLER

The Samsung Arrangeable Microcontroller (SAM) Assembler, SAMA, is a universal assembler, and generating an object code in the standard hexadecimal format. Assembled program codes include the object code used for ROM data and required In-circuit emulators program control data. To assemble programs, SAMA requires a source file and an auxiliary definition (device_name.def) file with device specific information.

HEX2ROM

HEX2ROM file generates a ROM code from a HEX file which is produced by the assembler. A ROM code is needed to fabricate a microcontroller which has a mask ROM. When generating a ROM code (.OBJ file) by HEX2ROM, the value "FF" is automatically filled into the unused ROM area, up to the maximum ROM size of the target device.

TARGET BOARDS

Target boards are available for all the S3C9-series microcontrollers. All the required target system cables and adapters are included with the device-specific target board. TB9484/88 is a specific target board for the S3C9484/C9488/F9488 development

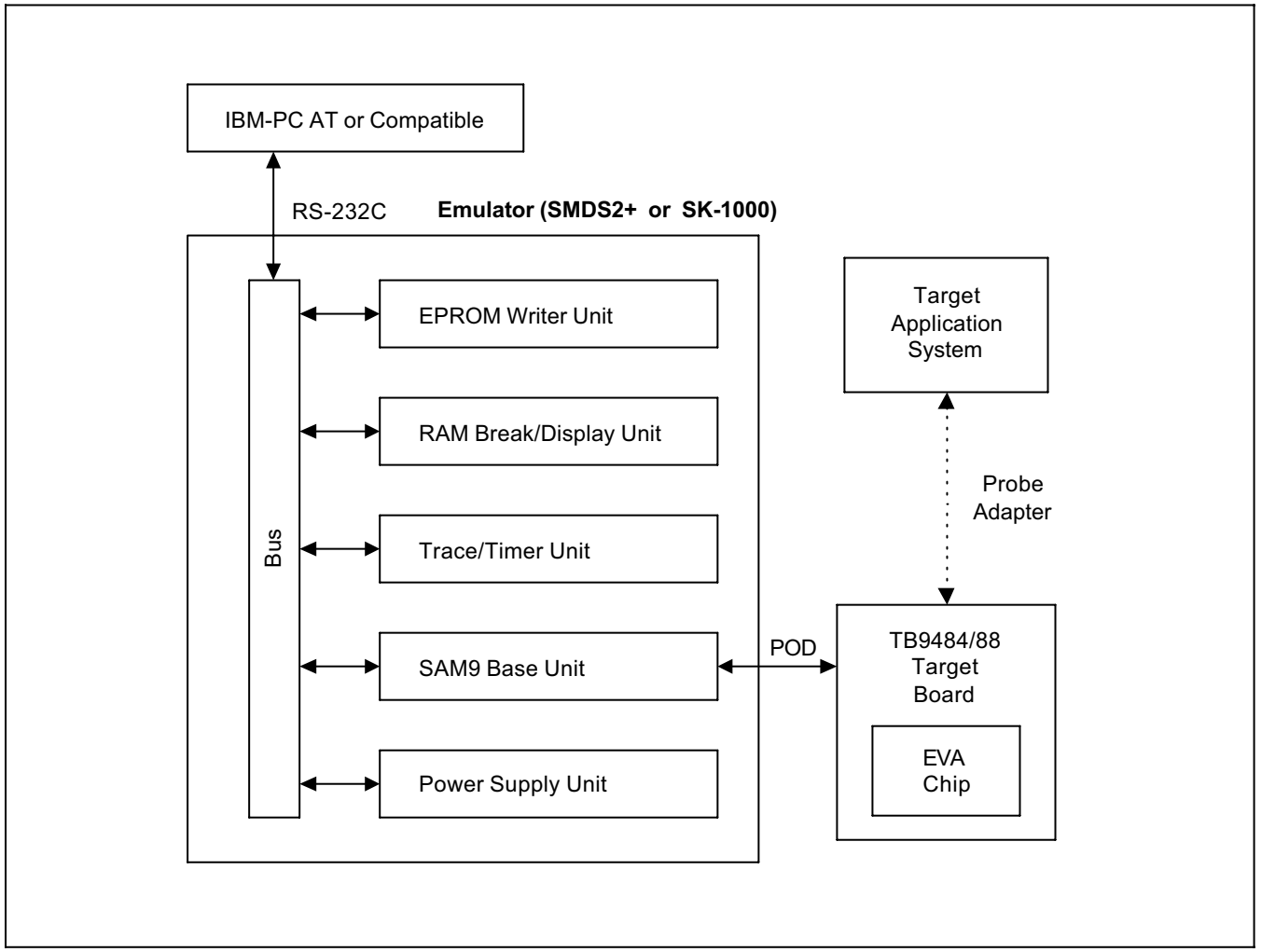


Figure 22-1. SMDS+ or SK-1000 Product Configuration

TB9484/9488 TARGET BOARD

The TB9484/9488 target board is used for the S3C9484/C9488/F9488 microcontrollers. It is supported by the SK-1000/SMDS2+ development systems.

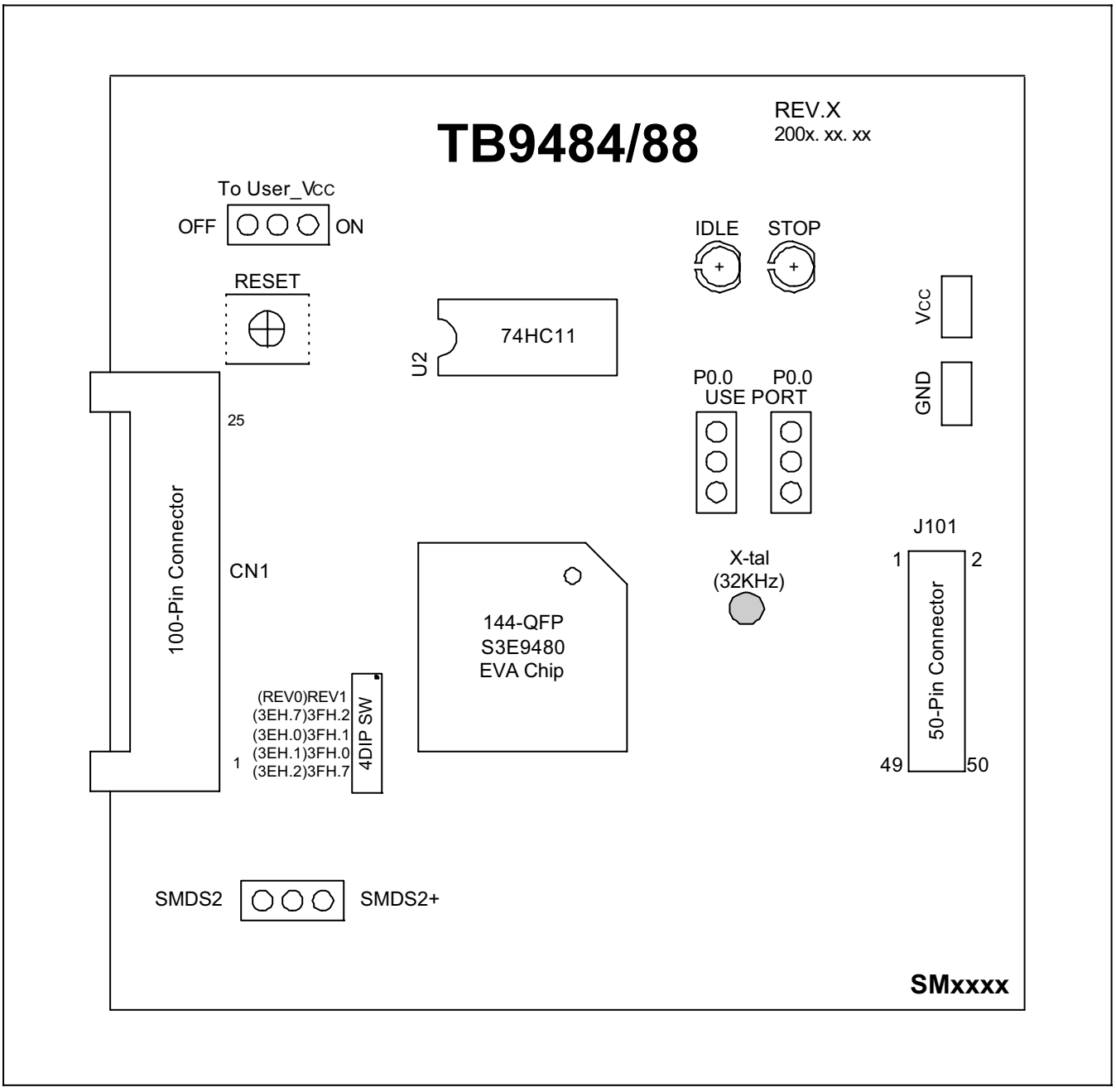
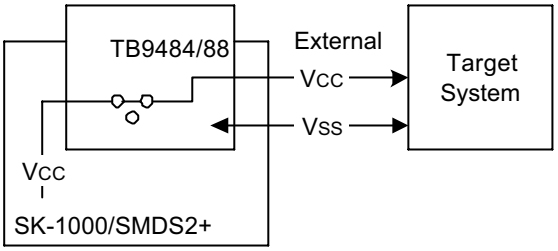
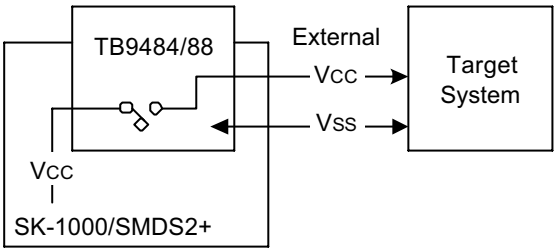


Figure 22-2. TB9484/88 Target Board Configuration

Table 22-1. Power Selection Settings for TB9484/88

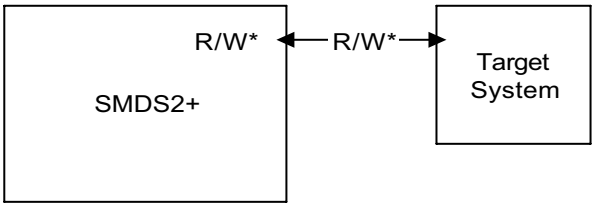
"To User_Vcc" Settings	Operating Mode	Comments
To user_Vcc off on		The SK-1000/SMDS2+ main board supplies V_{CC} to the target board (evaluation chip) and the target system.
To user_Vcc off on		The SK-1000/SMDS2+ main board supplies V_{CC} only to the target board (evaluation chip). The target system must have its own power supply.

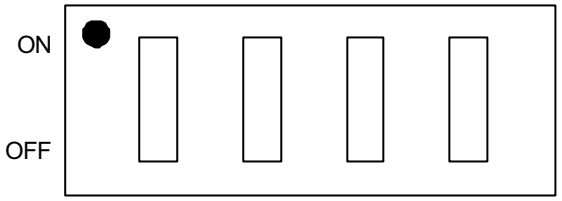
NOTE: The following symbol in the "To User_Vcc" Setting column indicates the electrical short (off) configuration:

SMDS2+ Selection (SAM8)

In order to write data into program memory that is available in SMDS2+, the target board should be selected to be for SMDS2+ through a switch as follows. Otherwise, the program memory writing function is not available.

Table 22-2. The SMDS2+ Tool Selection Setting

"SW1" Setting	Operating Mode
SMDS SMDS2+	



3FH.2 3FH.1 3FH.0 3EH.7 : Target board revision 1
3FH.7 3FH.0 3FH.1 3FH.2 : Target board revision 0

ON	Low
OFF	High

NOTES:
1. There is no ROM in the EVAchip. So smart option is not determined by software but DIP switch.
2. Target board revision number is printed on the target board (Refer to the Figure 22-2.)

Figure 22-4. DIP Switch for Smart Option

SWITCH	ON	OFF
3FH.2	XTin / XTout enable	Normal I/O pin enable
3FH.1	Internal RC oscillator	Basic Timer overflow used
3FH.0	Normal I/O pin enable	RESET Pin enable
3EH.7	LVR disable	LVR enable

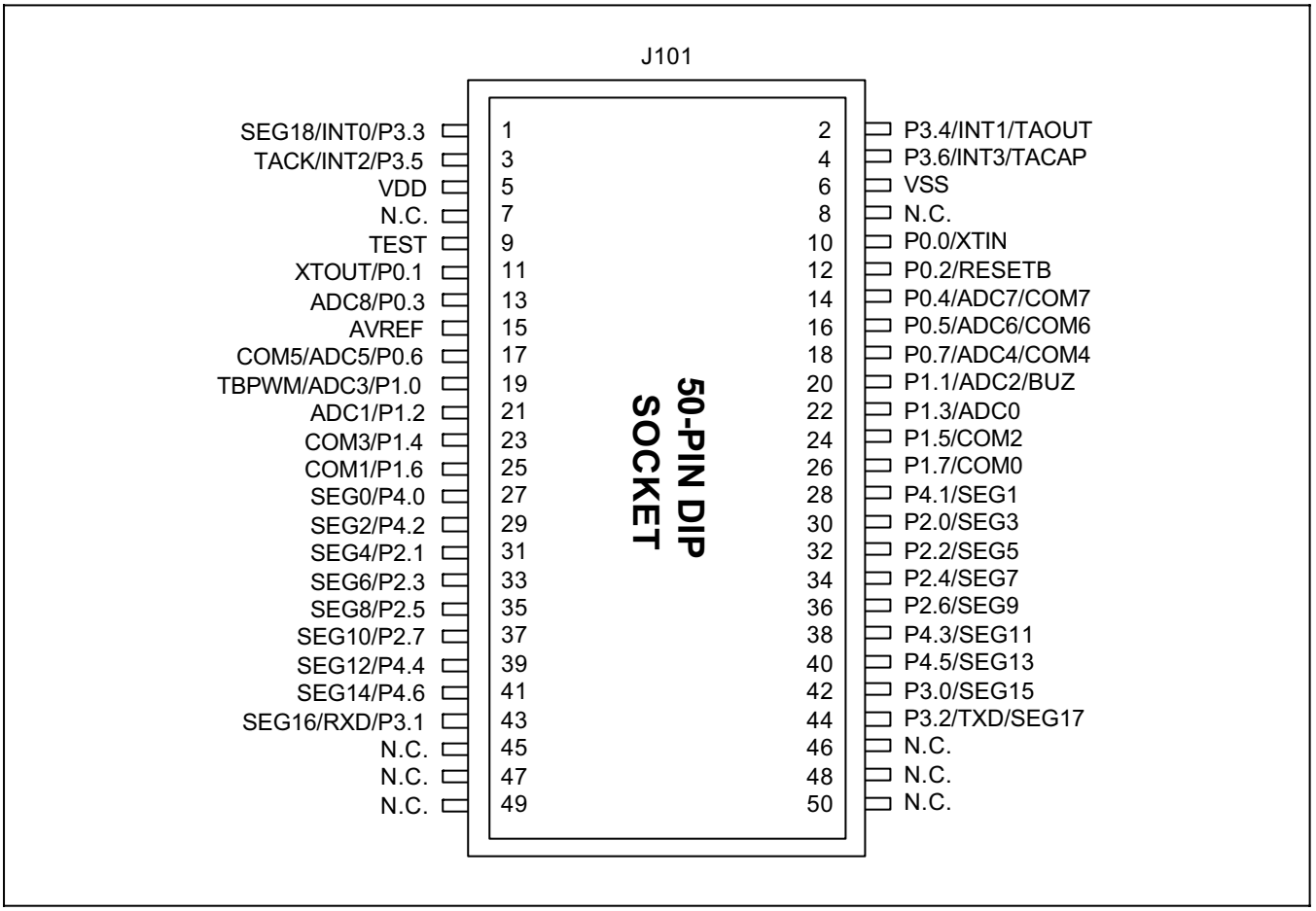


Figure 22-4. 44-Pin Connector for TB9484/88

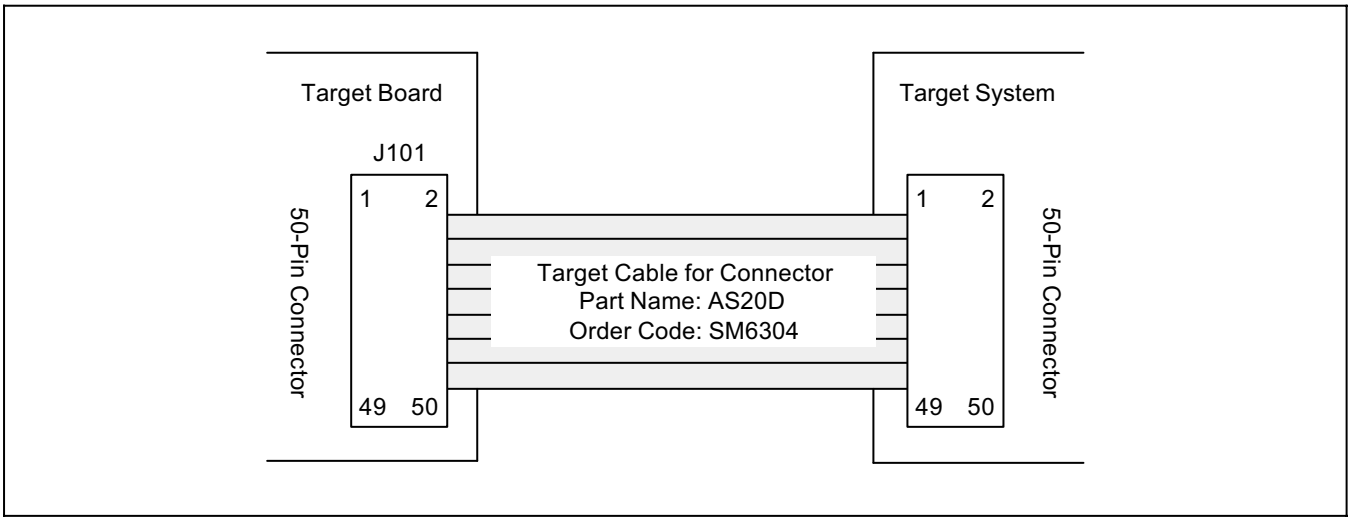


Figure 22-5. S3C9484/C9488/F9488 Probe Adapter for 44-pin Connector Package